pom's

La revue francophone des utilisateurs de l'Apple

- L'instruction "Library" du MSBasic
- Un éditeur graphique : Edigraph
- Conversion Pascal vers DOS
- Des disquettes à 36 pistes
- Défilement graphique
- RWTS désassemblée
- Du courrier en WPL
- Mac : un accessoire
- Calcul de π
- Décrypteur

NUMERO 21 . PRIX 40 F

M2366-21-40 F

ISSN : 0294-6068

l'actualité

les bancs d'essai

les guides d'achat

le dossier

les programmes

L'ORDINATEUR INDIVIDUEL

LA RÉFÉRENCE EN MICRO-INFORMATIQUE

A L'ORDINATEUR INDIVIDUEL**, les rédacteurs, les conseillers techniques, les correspondants à l'étranger, l'équipe entière se mobilise pour vous fournir tous les mois une information complète et de qualité. Le monde de la micro bouge : **L'O.I.** teste pour vous les micros et logiciels qui apparaissent sur le marché. Il vous dit lesquels choisir et pourquoi. Vous êtes déjà équipé et vous souhaitez tirer le maximum de votre machine ? Les spécialistes de **L'O.I.** vous livrent conseils, programmes inédits et astuces d'utilisation. Lisez chaque mois **L'ORDINATEUR INDIVIDUEL.

Conversion Pascal/Dos

Jean-Claude Lengrand

][+ //e //c

Page 6

Edigraph

Roland Jost

][+ //e //c

Page 9

Courrier automatisé en WPL

Véronique Schmitt

//e //c

Page 18

RWTS désassemblée

Patrice Neveu

Page 21

Décrypteur

Paul Fransceschi

][+ //e //c

Page 30

Library : un 'Motif' de satisfaction

Marianne Sutz

Page 40

Une corbeille dans la

Jean-Luc Bazanegue

Fenêtres de saisie

Gérard Michel

Page 47

La 36ème piste

B. Toméno

][+ //e //c

Page 51

Un compatible in compatible ?

Yvan Koenig

Page 53

Défilement graphique

Patrick Quettier

][+ //e //c

Page 59

Calcul de π

Michel Dominique

][+ //e //c

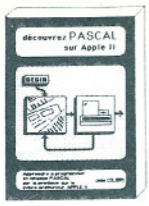
Page 65

Editorial	5
Bibliographie	67
Courrier des Lecteurs	69
Micro-informations	71
Entretien avec Jean-Louis Gassée	73

Les annonceurs :
A.C.I. : page 75. Apple : pages 38 et 39. I.E.F. : page 25. Mnémodyne : page 4.
Ordinateur Individuel : page 2. PSI : page 76. Télécompo : page 52.

Editions MEV - 64/70 rue des Chantiers - 78000 Versailles. Directeur de la publication : Hervé Thiriez

Pom's numéro 21 - Novembre-Décembre 1985



DECOUVREZ PASCAL SUR APPLE][, //e, //c

John COLIBRI
La nouvelle édition en 2 tomes de ce grand classique. Permet une initiation pas à pas, par l'exemple. Décrit les nouveautés de la version 1.2, et l'utilisation du PASCAL sur APPLE //e et APPLE //c. Comprend aussi les schémas mémoire et les exemples d'exécution.

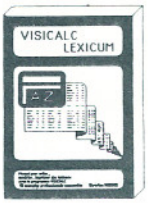
Tome 1: matériel nécessaire, branchements, amorçage, variables, contrôle du programme, procédures et fonctions (localité, paramètres, récursivité). 352 pages, 57 programme, 193 schémas et figures.
Le livre seul:.....148 F TTC
La disquette du livre.....75 F TTC

Tome 2: tableaux, enregistrements, fichiers, ensembles, pointeurs. Unités et segments, chaînage, fichiers de commande, lancement automatique. Une mine pour le traitement des fichiers, un exposé complet sur les pointeurs, et 33 fabuleuses annexes qui en font un ouvrage de référence. 424 pages, 175 figures, 63 programmes.
Le livre seul:.....148 F TTC
La disquette du livre.....75 F TTC

Disquette d'utilitaires (installation, impression de programmes, utilisation de la souris, conversions, etc)
La disquette d'utilitaires:..... 98 F TTC

BASIC APPLESOFT DOS3.3 et PRODOS

Paul MERRY
Guide de référence alphabétique. Définition, utilisation, exemples, types d'erreurs. Programmes de gestion de fichiers, calculs scientifiques, graphiques haute et basse résolution. 3ième édition augmentée avec PRODOS, 102 programmes, 641 pages.
Le livre seul.....168 F TTC
La disquette du livre.....75 F TTC



MAGICALC TUTORIAL

Christian B DUBOS
Manuel pas à pas pour apprendre à utiliser MAGICALC. Contient 10 tableaux professionnels commentés: devis, planning de fabrication, bulletin de salaire, facturation/stock, investissements, budgets, CEG/bilans, prévisions, stratégie d'entreprise, étude de prix. 383 pages
Le livre seul.....148 F TTC
La disquette du livre.....75 F TTC

RECUEIL PASCALISSIME

Recueil de programmes et d'articles PASCAL

Recueil 1: étiquettes - dump Epson - tri disque - modules assembleurs - séquentiel indexé - baguenaudier - lettres géantes - fichiers graphiques.

Recueil 2: labyrinthe - copie de fichiers - les routines réelles - source de l'interprète 6502 - perspective 3D - assembleurs 6502 symbolique.

Recueil 3: système expert - gestion d'adresse - générateur de menu - compilateur nano PASCAL - l'éditeur Elastique - utilisation de la souris - initiation PROLOG - tutorial Turbo PASCAL 1/N

Chaque recueil.....120 F TTC
Les disquettes de chaque recueil.....80 F TTC

TOPIQUES PASCAL

John COLIBRI (septembre 85)
Descriptions et utilisation des techniques avancées et du fonctionement système. Conversions de types - Entrées / sorties directes - Tableaux dynamiques - Interprète Visicalc - Fichiers de longueur variable - Fichiers Dynamiques - Implémentation: plan mémoire, Operating System, Interprète, Segments et Unités. 50 programmes. 400 pages.
Le livre seul.....290 F TTC
La disquette du livre.....98 F TTC

VISICALC TUTORIAL

Christian B DUBOS
Manuel pas à pas pour apprendre à utiliser VISICALC. Contient 10 tableaux professionnels commentés: devis, planning de fabrication, bulletin de salaire, facturation/stock, investissements, budgets, CEG/bilans, prévisions, stratégie d'entreprise, location de matériel. 375 pages
Le livre seul.....145 F TTC
La disquette du livre.....75 F TTC

VISICALC LEXICUM

Christian B DUBOS
Guide de référence alphabétique. Définition, utilisation, exemples pour les commandes, fonctions et manipulations VISICALC. Consolidation de tableaux, contrôle d'imprimante, interfaçage avec des programmes BASIC. 375 pages
Le livre seul.....145 F TTC
La disquette du livre.....75 F TTC

INITIATION PASCAL

John COLIBRI
Stage de formation avec le célèbre spécialiste PASCAL. Methode pas à pas, avec travaux pratiques sur IBM PC, Apple II, Mac Intosh. Conventions de formation.
Types de données simples, structures de contrôle, procédures et fonctions, données structurés, gestion de fichiers, tris en mémoire, programmation structurée.
Prochaines sessions: 7 Janvier, 4 Mars, 13 Mai, 1er Juillet, 5 Aout, 23 Septembre, 4 Novembre
4 jours..... **4900 F HT**

GESTION DE FICHIERS PASCAL

John COLIBRI
Stage de formation animé par un virtuose des fichiers. Travaux pratiques sur IBM PC, Apple II, Mac Intosh. Conventions de formation.
Saisie contrôlée, gestion de masques, structures de tri (hash code, B arbres) et séquentiel indexé. Programmes importants (compilation séparée, segmentation, chaînage). Fonctionnement de l'Operating System et utilitaires PASCAL. Communications. Compatibilité et transferts entre versions PASCAL.
Prochaines sessions: 3 Décembre, 4 Février, 8 Avril, 3 Juin, 7 Octobre, 2 Décembre
4 jours..... **4900 F HT**

DISQUETTES DE PROGRAMMES SOURCE

Textes sources. Format Apple 2)
- Le compilateur P4 de Zurich. Le langage PASCAL complet..... 190 F TTC
- Le compilateur / interprète PASCAL S (compilateur interpète en mémoire, sans les pointeurs et sans les fichiers)..... 145 F TTC
- Le compilateur PASCAL C (PASCAL S moins les réels, plus la concurrence)..... 90 F TTC
- L'interprète de P CODE 6502 (texte de l'interprète et du BIOS désassemblés)..... 190 F TTC
- PASCALISSIME 1..... 40 F TTC
- PASCALISSIME 2..... 40 F TTC
- PASCALISSIME 3..... 40 F TTC
- PASCALISSIME 4..... 40 F TTC
- PASCALISSIME 5..... 40 F TTC
- PASCALISSIME 6..... 40 F TTC
- Découvrez PASCAL Tome 1:..... 75 F TTC
- Découvrez PASCAL Tome 2:..... 75 F TTC
- Attach Bios (utilitaire de rattachement de périphériques)..... 98 F TTC

LE MAGAZINE PASCALISSIME

- **Le magazine consacré au langage PASCAL.** Progammes source commentés, présentation de nouvelles implémentations, compte-rendus de séminaires, initiation, bibliographie.
Au numéro:..... 25 F TTC
Abonnement de 12 numéros:..... 240 F TTC
Abonnement de 12 numéros et 4 disquettes (format Apple UCSD ou IBM MS DOS):.. 290 F TTC

LIBRAIRIE

Plus de 200 livres sur PASCAL. Ouvrages sur MODULA, Lisp, Prolog, Smalltalk, ADA. Collections américaines (Addison Wesley, Hayden, McGraw Hill, Osborne, Prentice Hall, Reston, Springer Verlag, Wiley) et françaises (Dunod, Eyrolles, Gauthier Villars, Masson, Sybex). Documentations et Magazines Pascal. Rayon PROLOG en plein essor.

GUIDEX

Cartes de référence en couleur. **L'essentiel à la portée de la main.**
GUIDEX ERREURS PASCAL (compilation, exécution, entrée-sortie) - GUIDEX APPLE PASCAL (périphériques, options de compilation, extensions) - GUIDEX SYNTAXE UCSD - GUIDEX CODES ASCII (table des codes, entrée des codes, fonctions système)
Chaque GUIDEX:..... 19 F TTC

CLUB DES UTILISATEURS PASCAL

SESSIONS INITIATION AVEC TRAVAUX PRATIQUES: Tous les 15 jours, le mercredi de 19h30 à 21h. 13 Novembre: procédures et fonctions.
SESSIONS PRESENTATIONS: tous les 15 jours, le mercredi de 19h30 à 21h30. Versions PASCAL, utilitaires et outils, PASCAL et les langages LISP, PROLOG, PILOTE, communications, traitements graphiques etc. 4 Déc: MAC PASCAL, 18 Déc: arbres binaires à clés de taille variable, 15 Janv: bibliographie à l'aide de fichiers semi structurés. Les minutes des présentations sont publiées dans Pascalissime.
Cotisation annuelle:..... **290 F TTC**

OUTILS 8088 / IBM PC

- **ASSEMBLEUR SYMBOLIQUE 8086**
Edition de texte, assemblage et exécution en mémoire. Peut générer un fichier .COM ou du texte Inline Turbo..... 490 F TTC
- **DESASSEMBLEUR SYMBOLIQUE 8086**
Fichier de paramètres des zones de données. Validation des labels. Re définition des labels. Sortie écran ou fichier ré assemblable 490 F TTC
- **DESASSEMBLAGE DE TURBO PASCAL**
Versin 2 ou 3 (nécessite Turbo.com, non fourni)..... 290 F TTC

LES LANGAGES EVOLUES

- **PASCAL UCSD** Environnement de développement
Version IV IBM PC sous MS DOS: 5950 F TTC
Version 1.2 Apple (utilise 128K //e) 1500 F TTC
- Compilateur **MODULA II** - Génère du code natif
Version Lisa (Mac Intosh en Mai).. 2900 F TTC
- **R LISP 2.0** (le MAC LISP de Winston intégral)
Disquette Apple // 128K, et manuel.. 590 F TTC

PROLOG. 1.3

Système de développement PROLOG complet. Destiné essentiellement à maîtriser rapidement le langage et à comprendre ses domaines d'utilisation.
Editeur de règles pleine page. Interprétation en mémoire centrale. Possibilité de trace hiérarchique mise en oeuvre à n'importe quel moment. Explication des résultats. Mesure de l'efficacité de la solution. Syntaxe simple. Version APPLE 2 et IBM PC.
Le système PROLOG 1.3:..... **490 F TTC**

PRODUITS PASCAL APPLE 2

- **MNEMO DUMP**: affichage du contenu d'une disquette ou d'un fichier, affichage du contenu de la mémoire..... 90 F TTC
- **PERUSE**: édition rapide du contenu de fichiers texte. Facilite la recherche de textes... 95 F TTC
- **PIP.EDITOR**: éditeur UCSD amélioré (table des matières, impression ligne à ligne, généralisation de P(age et A(djust etc)..... 190 F TTC
- **MNEMO DE PCODE** désassemblage d'un fichier compilé. Mnémoniques en toutes lettre.. 290 F TTC
- **Transfert de fichiers Apple** UCSD -> DOS 3.3 et DOS 3.3 -> UCSD..... 290 F TTC
- **Communicatons Apple UCSD/IBM UCSD**.. 490 F TTC
- **Communicatons Apple UCSD/IBM TURBO** 490 F TTC

Vous pouvez acheter ces produits directement, ou les commander par correspondance en envoyant votre commande et le chèque du montant à:

MNEMODYNE
26 rue Lamartine
75009 Paris

Tel 42 85 10 82 - Métro Cadet
9h-12h 13h-18h, Mercredi jusqu'à 19h30
Fermé le samedi

Ces produits seront visibles sur le stand de **L'INSTITUT PASCAL** lors du **SICOB BOUTIQUE**, du **SICOB DE PRINTEMPS** ou d'**APPLE EXPO**.

Je commande:
PRODUIT: PRIX TTC:

_____ _____
_____ _____
_____ _____

Le port en France Métropolitaine est compris. Pour les autres destinations: ajoutez 20F par produit. Précisez le format des disquettes commandées.

NOM et ADRESSE:

Editorial

La presse nous inonde depuis deux mois d'articles sur les péripéties d'Apple, avec le départ de Jobs qui a suivi celui de Wozniak. Les financiers ne semblent pas se fier aux nombreux bruits alarmistes : l'action d'Apple flotte aujourd'hui (fin octobre) aux environs de $18, soit 10% de mieux que la moyenne du cours ces derniers mois.

Cela tient aussi probablement au succès obtenu par Apple contre Digital Research au sujet du système GEM pour IBM; ce système imitait par trop la présentation et les graphismes déjà devenus classiques grâce au Macintosh : Digital Research est obligé de retirer GEM du marché le 15 novembre 85, doit verser de l'argent à Apple ("undisclosed amount") et soumettre ses nouvelles versions à Apple avant de les sortir sur le marché. Voilà qui ralentira la sortie effective de l'Atari 520 ST qui, avec l'Amiga de Commodore, est un des deux concurrents du Macintosh; en effet, le 520 ST devait lui aussi utiliser le système GEM.

N'oublions pas non plus deux faits importants : tout d'abord, la gamme Apple s'enrichit de jour en jour, qu'il s'agisse de la ligne Apple II ou de la ligne Macintosh; et enfin, quel est le constructeur de micros qui ne connaît pas de problème ces temps-ci ? Même IBM France a été obligé de revoir ses prévisions de vente d'IBM PC pour l'année 1985, en baisse de 100 000 à 60 000 unités ! Nous gardons donc notre moral et continuons à vous inonder d'articles et de programmes, dont nous nous efforçons de maintenir le niveau de qualité au plus haut.

Voici pour ce numéro un programme chargé : un décrypteur de ***Paul Franceschi*** *bien pratique pour ne pas rester bloquer dans les jeux d'aventures. Sans arriver aux 11 millions de décimales déjà connues, vous calculerez π avec* ***Michel Dominique****.*

Côté graphique, deux contributions : ***Patrick Quettier*** *et* ***Roland Jost****, pour rêver en HGR... Etudes théoriques :* ***Yvan Koenig*** *s'est penché sur les ROMs de la ligne Apple et* ***Patrice Neveu*** *a désassemblé RWTS.*

Théorique aussi et d'une utilisation immédiate les 36 pistes de ***B. Toméno****. Plus rare, une contribution en WPL de* ***Véronique Schmitt****.*

Enfin, pour convertir vos fichiers Pascal en des fichiers DOS, "Lirpastext" de ***Jean-Claude Lengrand****.*

Nouvelles progressions sur le Macintosh : ***Jean-Luc Bazanegue*** *a mis au point un accessoire qui devient vite indispensable : enfin pouvoir effacer des fichiers depuis une application...* ***Marianne Sutz****, également une habituée de nos colonnes, vous présente une instruction méconnue du MSBasic 2.0 : "Library". Côté pédagogie, nous retrouvons* ***Gérard Michel*** *avec le fonctionnement de fenêtres de saisie.*

Bien sûr vous retrouverez nos rubriques habituelles avec ***Alexandre Duback, Jean-Michel Gourévitch, Alexandre Avrane*** *et* ***Olivier Herz****.*

Hervé Thiriez

Photo de couverture : nouvelle imprimante Apple ImageWriter][

Ont collaboré à ce numéro : Alexandre Avrane, Jean-Luc Bazanegue, Michel Dominique, Alexandre Duback, Paul Franceschi, Jean-Michel Gourévitch, Olivier Herz, Roland Jost, Yvan Koenig, Jean-Claude Lengrand, Gérard Michel, Patrice Neveu, Patrick Quettier, Véronique Schmitt, Marianne Sutz, B. Toméno.
Directeur de la publication, rédacteur en chef : Hervé Thiriez.
Rédacteurs : Alexandre Avrane, Olivier Herz.
Siège social : Editions MEV - 49, rue Lamartine - 78000 Versailles. Tél. : (1) 39.51.24.43.
Publicité : Editions MEV.
Diffusion : N.M.P.P.
Impression : Rosay - 47, avenue de Paris - 94300 Vincennes. Tél. : (1) 43.28.18.63.

Conversion Pascal DOS

Jean-Claude Lengrand

Transformer des fichiers Pascal en des fichiers DOS, tel est l'objet du programme présenté ici : *Lirpastext*. Les fichiers PASCAL en question peuvent être des FILE OF CHAR (TEXT au sens du langage PASCAL) de deux types différents (selon qu'ils ont ou non été créés par l'éditeur, c'est-à-dire qu'ils ont ou non le suffixe .TEXT) ou des FILE OF STRING. Les premiers sont convertis en fichiers DOS 3.3 à accès séquentiel, les seconds en fichiers DOS 3.3 à accès direct. Au passage, certaines conversions de caractères peuvent être effectuées (remplacement des virgules par des blancs, par exemple). Pour utiliser *Lirpastext*, il faut connaître le numéro de bloc de début et la longueur du fichier sur la disquette PASCAL, ce que l'on voit immédiatement sur le catalogue en Pascal. Les messages du programme Basic en constituent le mode d'emploi.

Programme LIRPASTEXT

```
10 REM LIRPASTEXT
20 REM RECOPIE UN "FILE OF CHAR" PASCAL
30 REM OU UN FILE OF STRING PASCAL
40 REM SUR UN FICHIER SEQUENTIEL BASIC (
   FILE OF CHAR)
50 REM OU A ACCES DIRECT BASIC (FILE OF
   STRING)
60 REM LENGRAND,  LE 21/12/82
70 REM
80 HIMEM: 38090
90 AD = 38091
100 DQ$ =  CHR$ (4):BI$ =  CHR$ (7):DL$
    =  CHR$ (16)
110 DIM S(15),S1(2)
120 FOR I = 0 TO 15: READ S(I): NEXT I
130 REM  LISTE DES SECTEURS BASIC CORRES
    PONDANT
140 REM  AUX SECTEURS PASCAL 0,1,2,...
150 DATA  0,14,13,12,11,10,9,8,7,6,5,4,3
    ,2,1,15
160 BU = AD + 21: REM  ADRESSE DU BUFFER
170 PG = AD + 279: REM  ADRESSE DU PROGR
    AMME DE LECTURE
180 ER = AD + 13: REM  AD.DU CODE ERREUR
190 TR = AD + 4: REM  AD. DU NUMERO DE P
    ISTE
200 SC = AD + 5: REM  AD. DU NUMERO DE S
    ECTEUR
210 OD = AD + 16: REM  AD. DU DERNIER DR
    IVE UTILISE
220 ND = AD + 2: REM  AD.DU FUTUR DRIVE
230 S = 0: DIM S1%(10),S2%(10): REM  POU
    R LA SUBSTITUTION DE CARACTERES
240 TEXT : HOME : PRINT "TRANSFERT D'UN
    FICHIER DE CHARACTERES"
250 PRINT "PASCAL SUR UNE DISQUETTE BASI
    C": PRINT
260 PRINT "METTEZ LA DISQUETTE CONTENANT
     USERWTS"
270 PRINT "DANS LE DRIVE 1 ET FAITES <RE
    TURN>"
280 GET W$: PRINT : PRINT
290 PRINT DQ$;"BLOAD USERWTS.CODE,A";AD;
    ",D1"
300 PRINT "METTEZ LA DISQUETTE PASCAL A
    LIRE"
310 PRINT "DANS LE DRIVE 2 ET FAITES <RE
    TURN>"
320 GET W$: PRINT : PRINT
330 PRINT "METTEZ LA DISQUETTE BASIC DES
    TINATAIRE"
340 PRINT "DANS LE DRIVE 1 ET FAITES <RE
    TURN>"
350 GET W$: PRINT : PRINT
360 HTAB 1: VTAB 4: CALL  - 958
370 INPUT "PREMIER BLOC A LIRE=";PB
380 IF PB < 0 OR PB > 279 THEN  PRINT BI
    $: GOTO 370
390 INPUT "NOMBRE DE BLOCS A LIRE=";NB
400 IF NB < 0 OR NB > 279 THEN  PRINT BI
    $: GOTO 390
410 INPUT "EST-CE UN FILE OF STRING (O/N
    ) ?";S$
420 IF  LEFT$ (S$,1) = "N" THEN S$ = "N"
    : PRINT "CE DOIT ETRE UN FILE OF CHA
    R": GOTO 480
430 IF  LEFT$ (S$,1) <  > "O" THEN  PRIN
    T BI$: GOTO 410
440 S$ = "O":T$ = "N":N% = 1
450 INPUT "LONGUEUR DECLAREE POUR LES CH
    AINES ? ";LM
460 GOTO 570: REM  CE NE PEUT PAS ETRE U
    N .TEXT
480 INPUT "EST-CE UN FICHIER .TEXT (O/N)
     ? ";T$
490 IF  LEFT$ (T$,1) = "O" THEN T$ = "O"
    : GOTO 520
500 IF  LEFT$ (T$,1) = "N" THEN T$ = "N"
```

```
: GOTO 520
510 PRINT BI$: GOTO 480
520 PRINT : PRINT "VOULEZ-VOUS    (1 OU 2
    ):"
530 PRINT "1 RECOPIER TOUS LES CARACTERE
    S MEME NULS"
540 PRINT "2 VOUS ARRETER AU PREMIER CAR
    ACTERE NUL"
550 PRINT : PRINT "NB:CARACTERE NUL ($00
    )= FIN DE FICHIER"
560 INPUT "";N%: IF N% <  > 1 AND N% <
    > 2 THEN  PRINT BI$;: GOTO 560
570 HOME : PRINT "A L'OCCASION DU TRANSF
    ERT, VOUS"
580 PRINT "POUVEZ REMPLACER CERTAINS CAR
    ACTERES"
590 PRINT "PAR D'AUTRES (MAXI:10 SUBSTIT
    UTIONS)"
600 : PRINT "NBRE DE SUBSTITUTIONS DEMAN
    DEES ?"
610 PRINT "(PAR DEFAUT, LES ";S;" SUBSTI
    TUTIONS"
620 PRINT "PRECEDENTES)"
630 PRINT : INPUT "INDIQUEZ LE NOMBRE OU
     <RETURN>";W$
640 IF W$ = "" GOTO 750
650 W =  VAL (W$): IF W < 0 OR W > 10 TH
    EN  PRINT BI$: GOTO 630
660 S = W: IF S = 0 GOTO 810
670 FOR I = 1 TO S
680 HOME : PRINT "POUR LE CARACTERE ";I;
    " :"
690 INPUT "CODE ASCII DECIMAL INITIAL=";
    S1%(I)
700 PRINT "CODE ASCII DECIMAL FINAL :"
710 INPUT "(OU -1 POUR ELIMINER LE CARAC
    TERE)";S2%(I)
720 IF S1%(I) < 0 OR S1%(I) > 127 THEN
    PRINT BI$: GOTO 680
730 IF S2%(I) <  - 1 OR S2%(I) > 127 THE
    N  PRINT BI$: GOTO 680
735 IF (S2%(I) =  - 1) AND (S$ = "O") TH
    EN  PRINT BI$: GOTO 680
740 NEXT I
750 HOME : PRINT "RAPPEL DES SUBSTITUTIO
    NS:": PRINT
760 IF S = 0 THEN  PRINT "AUCUNE SUBSTIT
    UTION": PRINT : GOTO 810
770 FOR I = 1 TO S
780 PRINT "CHR$(";S1%(I);")=";  CHR$ (S1%
    (I));
790 PRINT " -> CHR$(";S2%(I);")=";  CHR$
    (S2%(I))
800 NEXT I: PRINT
810 PRINT "NOM DU FICHIER BASIC A CREER:
    "
820 INPUT "    ";FI$
830 IF S$ = "N" GOTO 860
840 PRINT "CE FICHIER DEVRA ETRE EXPLOIT
    E AVEC UN"
850 PRINT "RECORD LENGTH=";LM + 2;" (ACC
    ES DIRECT)"
860 PRINT : PRINT "<ESC> POUR CORRIGER U
    NE ERREUR"
870 PRINT "<RETURN> POUR LANCER LA COPIE
    "
880 GET W$: IF W$ <  >  CHR$ (27) AND W$
     <  >  CHR$ (13) THEN  PRINT BI$: GO
    TO 880
890 IF W$ =  CHR$ (27) THEN  HOME : GOTO
     360
900 IF T$ = "O" THEN PB = PB + 2:NB = NB
     - 2: REM  SUPPRIME LES 2 PREMIERS B
    LOCS
910 POKE AD + 12,1: REM  FONCTION LECTUR
    E
920 LA = 13: REM  ASCII DU DERNIER CARAC
    TERE LU
930 NS = 0: REM  COMPTAGE DES CARACTERES
     POUR LES FILE OF STRING
940 HOME : PRINT "VOUS ALLEZ VOIR DEFILE
    R CE QUI"
950 PRINT "S'ENREGISTRERA SUR VOTRE DISQ
    UETTE BASIC": PRINT
960 PRINT : PRINT DQ$;"MON O"
970 PRINT : PRINT DQ$;"OPEN ";FI$;",D1"
980 PRINT DQ$;"WRITE ";FI$
990 FOR I = 1 TO NB
1000 BL = PB + I - 1: REM   NUMERO DE BL
     OC
1010 PI =  INT (BL / 8): REM  NUMERO DE
     PISTE
1020 S1(1) = 2 * (BL - 8 * PI):S1(2) = S
     1(1) + 1: REM   NUMERO DES SECTEURS
      PASCAL
1030 POKE TR,PI: REM  AFFICHE LA PISTE
1040 FOR J = 1 TO 2: REM  POUR LES 2 PAR
     TIES DU BLOC
1050 ST = S(S1(J))
1060 POKE SC,ST: REM  AFFICHE LE SECTEUR
1070 POKE OD,1: REM  AFFICHE LE DERNIER
     DRIVE
1080 POKE ND,2: REM  AFFICHE LE DRIVE A
     UTILISER
1090 CALL PG: REM  APPELLE USERWTS
1100 IF  PEEK (ER) = 0 GOTO 1140
1110 PRINT DQ$;"CLOSE ";FI$
1120 PRINT "ERREUR DE LECTURE NUMERO ";
     PEEK (ER)
1130 GOTO 1410
1140 FOR K = 0 TO 255
1150 IF ( PEEK (BU + K) <  > 16) OR (T$
     = "N") OR (LA <  > 13) OR (S$ = "O"
     ) GOTO 1190
1160 NI =  PEEK (BU + K + 1) - 32
1170 IF NI > 0 THEN  FOR II = 1 TO NI: P
     RINT " ";: NEXT II
1180 K = K + 2
```

```
1190 LA =  PEEK (BU + K):LB = LA
1200 IF LA = 0 AND N% = 2 THEN K = 255:J
     = 2:I = NB: GOTO 1340
1210 IF S = 0 OR (S$ = "O" AND (NS = 0 O
     R NS > LS)) GOTO 1260
1220 FOR L = 1 TO S
1230 IF LA < > S1%(L) GOTO 1250
1240 LB = S2%(L):L = S
1250 NEXT L
1260 IF S$ = "O" GOTO 1290
1270 IF LB >  - 1 THEN  PRINT  CHR$ (LB)
     ;
1280 GOTO 1340
1290 IF NS = 0 THEN LS = LB: GOTO 1310:
     REM  LONGUEUR EFFECTIVE DE LA CHAIN
     E
1300 IF NS <  = LS THEN  PRINT  CHR$ (LB
     );
1310 IF NS = LS THEN  PRINT : GOTO 1320:
      REM  ENVOI D'UN RETURN
1315 IF NS > LS THEN  PRINT  CHR$ (0);
1320 NS = NS + 1
1330 IF NS > LM + 1 THEN NS = 0: REM   D
     EBUT D'UNE NOUVELLE CHAINE
1340 NEXT K
1350 NEXT J
1360 NEXT I
1370 PRINT
1380 PRINT DQ$;"CLOSE ";FI$
1390 PRINT DQ$;"NOMON O"
1400 PRINT : PRINT : PRINT BI$;BI$;BI$;"
     C'EST FINI.."
1410 PRINT : PRINT "<RETURN> POUR COPIER
      UN AUTRE FICHIER"
1420 PRINT "OU METTEZ LA DISQUETTE UTILI
     TAIRE"
1430 PRINT "   DANS LE DRIVE 1 ET FAITES
      <ESC>"
1440 PRINT "   POUR REVENIR AU MENU GENE
     RAL"
1450 PRINT "OU UNE AUTRE TOUCHE POUR ARR
     ETER"
1460 GET W$: PRINT
1470 IF W$ =  CHR$ (13) THEN  HOME : GOT
     O 300
1480 IF W$ =  CHR$ (27) THEN  PRINT DQ$;
     "RUN MENU,D1"
1490 END
```

Source USERWTS.ASS
(Assembleur LISA 1.5)

```
1          ORG  !38091
2          OBJ  $1000
3   ;MODE D'EMPLOI:
4   ;FAIRE LES MODIFICATIONS NECESSAIRES A
    LA TABLE IOB:
5   ;      NUM. DE PISTE ET DE SECTEUR
6   ;      CODE DE L'OPERATION A EFFECTUER
7   ;ET LANCER L'EXECUTION A L'ADRESSE PGM.
8   ;
9   IOB     HEX  01        ;IMPOSE
10  IBSLOT  HEX  60        ;16*SLOT 6
11  IBDRVN  HEX  01        ;DRIVE 1
12  IBVOL   HEX  00        ;NUMERO DE VOLUME
13  IBTRK   HEX  00        ;NUMERO DE PISTE
14  IBSECT  HEX  00        ;NUMERO DE SECTEUR
15  IBDCTP  ADR  IOB+17    ;ADRESSE DE LA DCT
16  IBBUFF  ADR  IOB+21
17          ADR  $0000     ;INUTILISE
18  IBCMD   HEX  00        ;CODE COMMANDE
19  IBSTAT  HEX  00        ;CODE ERREUR
20  IBSMOD  HEX  00        ;NUMERO DE LA
                           DISQUETTE EFFECTIVE
21  IOBPSN  HEX  60        ;DERNIER SLOT UTILISE
22  IOBPDN  HEX  01        ;ET DERNIER DRIVE
23  DEVTPC  HEX  0001EFD   ;4 BYTES IMPOSES
24  BUFFER  HEX  00        ;1ER OCTET DU BUFFER
25          DFS  255       ;LES 255 SUIVANTS
26  HADR    HBY  IOB
27  LADR    BYT  IOB
28  PGM     LDA  HADR
29          LDY  LADR
30          JSR  $03D9     ;APPELLE RWTS
31          LDA  IBSLOT    ;MAJ DERNIER SLOT
32          STA  IOBPSN    ;ET DERNIER DRIVE
33          LDA  IBDRVN
34          STA  IOBPDN
35          BCS  NOTOK
36          LDA  #!0       ;MET 0 SI OK
37          STA  IBSTAT
38  NOTOK   RTS
39          BRK
40          PAU
41          DCM  "BSAVE USERWTS.CODE,
                 A$1000,L$135"
42  END
```

Récapitulation USERWTS.CODE

```
1000- 01 60 01 00 00 00 DC 94 E0 94 00 00 00 00 00 60
1010- 01 00 01 EF D8 00 00 00 00 00 00 00 00 00 00 00
1020- 08 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1030- 00 00 00 00 00 00 00 00 00 00 08 00 00 00 00 00
1040- 00 00 00 00 00 02 00 00 00 00 08 00 00 10 00 10
1050- 00 00 00 00 00 00 88 80 00 00 00 80 00 00 00 80
1060- 00 00 20 00 10 00 00 00 00 08 00 00 00 00 80 80
1070- 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00 00
1080- FF FF D7 FF FF FF DF FF FF FF FF FF FF FF FF FF
1090- FF EF FF FF FF DF EF FF FF FF DF FF EF FF FF FF
10A0- FF DF EF FF FF FF FF FF F7 FF FF FF FF FF FF FF
10B0- FF FF FF FF FF FF FF FF FF FF FF FF FF FF EF FF
10C0- EF FF FF FF FF FF FF F7 FF FF FF FF FF FF FF FF
10D0- DF FF FF FF FF FF EF FF FF FF FF FF FF FF FF FF
10E0- FF FF FF FF FF FF DF FF 7F FF FF FF FF FF FF FF
10F0- DF FF FF FF FF FF FF FF FF FF FF FF FF FF EF FF
1100- 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1110- 00 00 00 00 00 94 CB AD E0 95 AC E1 95 20 D9 03
1120- AD CC 94 8D DA 94 AD CD 94 8D DB 94 B0 05 A9 00
1130- 8D D8 94 60 00
```

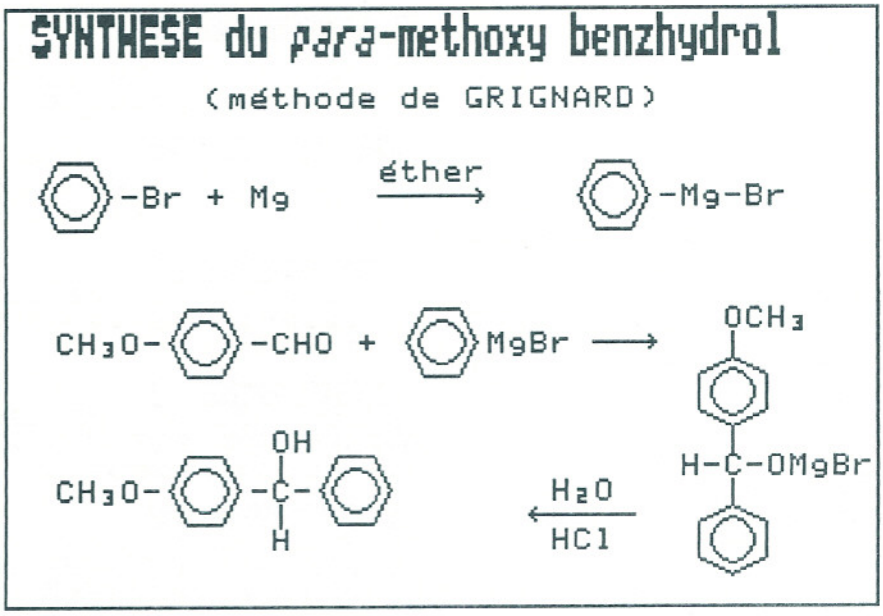
Un éditeur HGR : *Edigraph*

Roland Jost

Les traitements de texte classiques ne sont pas prévus pour l'écriture de textes scientifiques; il est à la rigueur possible d'écrire des formules mathématiques avec indices et exposants. Par contre, on ne peut dessiner facilement des formules chimiques ou des schémas électroniques, ou encore rajouter des commentaires sur des pages graphiques .
C'est pourquoi nous avons écrit *EDIGRAPH*. Nous aurions pu utiliser HRCG de TOOLKIT ou HAIFA (Pom's 5) ou HGRECR (Pom's 15), mais nous avons préféré écrire un programme entièrement en Basic pour mieux faire comprendre la gestion du graphisme Apple et permettre des modifications aisées par l'utilisateur final.

EDIGRAPH est loin d'avoir les possibilités classiques d'un traitement de texte, mais il permet par contre :

- l'écriture de textes et de formes en page graphique;
- le passage aisé d'un type de caractères à un autre;
- l'affichage de caractères en mode inverse, en double hauteur ou soulignés;
- caractères en indice obtenus par CTRL-I ou bouton de manette 0 (Apple][), ou pomme ouverte ou TAB (Apple //e);
- caractères en exposant obtenus par CTRL-X ou bouton de manette 1 (Apple][), ou pomme fermée (Apple //e);
- sauvegarde et rappel d'une page graphique;
- manipulation de zones écran; *EDIGRAPH* permet :
  1) de déplacer ou recopier une zone écran;
  2) de mettre en mode inverse une partie de l'écran;
  3) d'effacer une zone-écran;
  4) de stocker une zone-écran dans un tampon d'où elle pourra être rappelée à volonté. Le tampon pourra être sauvé sur disquette.
- adaptation aisée aux besoins propres d'un utilisateur , car étant écrit en Basic, les modifications logicielles sont simples. On peut aussi modifier les polices de caractères ou rajouter de nouvelles formes.



Affichage de caractères

Gestion d'une page graphique

La page graphique de l'Apple est divisée en 192 lignes et 280 colonnes. Les colonnes sont regroupées par 7, chaque point correspondant à un des 7 bits d'un octet, dont le 8ème bit contient le code couleur. L'écran HGR est divisé en 3 zones de 8 groupes de 8 lignes. Par analogie avec la page TEXTE, la page graphique peut donc être considérée comme comportant 964 cases (24 lignes * 40 colonnes) de 7 * 8 points élémentaires. On pourra inscrire un caractère dans chaque case.

Codage d'un caractère

Le caractère est dessiné dans une matrice 8 * 8 , puis le codage est fait ligne par ligne. Par exemple, la lettre B sera codée comme suit sur huit octets.

	binaire	hex
oct.1	○○○●●●●○	1E
oct.2	○○●○○○●○	22
oct.3	○○●○○○●○	22
oct.4	○○○●●●●○	1E
oct.5	○○●○○○●○	22
oct.6	○○●○○○●○	22
oct.7	○○○●●●●○	1E
oct.8	○○○○○○○○	00

vous remarquerez que la lettre est inversée.

Ecriture d'un caractère en page HGR

Pour écrire un caractère, il faut afficher le contenu des 8 octets

qui servent à le définir les uns sous les autres, à l'endroit voulu. Il faut donc poker dans les bonnes adresses de la page mémoire.
L'adresse du sommet gauche d'une case est obtenue par la formule :

*BA = Z2 + 128 * ZV - 984 * INT ((ZV-1)/8) + ZH* avec :

Z2 = 8127 pour la page 1, 16255 pour la page 2

ZV = position verticale (0 <= ZV <= 23)

ZH = position horizontale (0 <= ZH <= 39)

L'écriture en mémoire HGR des huit octets du caractère se fait par la boucle :

```
FOR J = 0 TO 7: POKE BA + 1024 * J, PEEK (ZB + ZC * 8 +J) : NEXT
```

ZB + ZC * 8 correspond à la position du caractère dans la table chargée à l'adresse ZB.

Pour un affichage en mode inverse, on effectue au préalable un AND sur chaque octet:. Pour celà, faire 255 - PEEK (ZB+ZC * 8+J).
Pour un affichage en double hauteur, on affiche deux fois verticalement le même octet.

Les jeux de caractères

Les caractères utilisés par *EDIGRAPH* sont semblables à ceux du Toolkit ou de HAIFA. Cinq polices de caractères sont stockées dans le fichier binaire SET chargé à l'adresse $3000 (12288) :

$3000-$32FF : caractères français (avec les minuscules accentuées)

$3300-$35FF : caractères grecs et symboles mathématiques (racine, intégrale,...) et chimiques (liaisons simple, double, triple, flèches de réaction, etc.)

$3600-$38FF : caractères réduits en matrice 3*5 pour les indices et exposants

$3900-$3BFF caractères larges (avec minuscules accentuées)

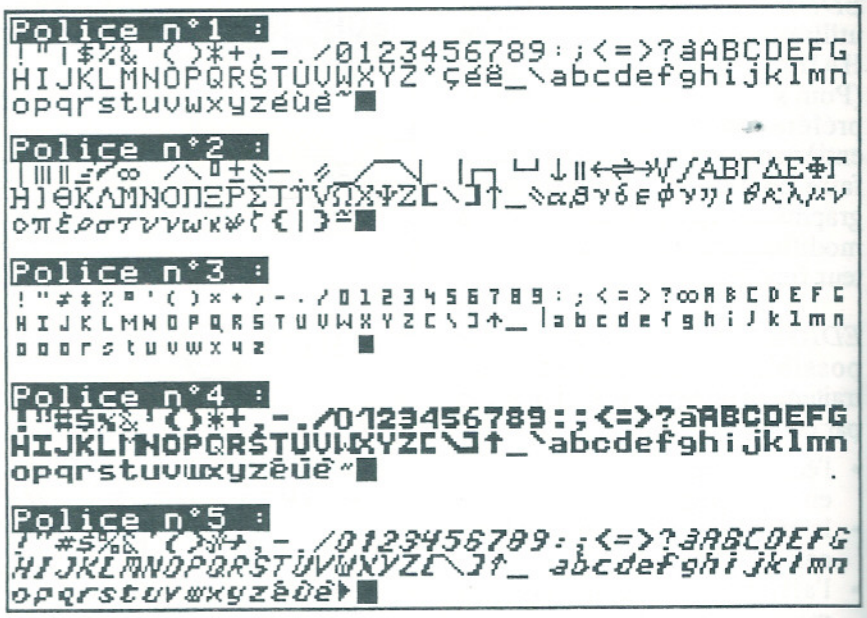
$3C00-$3EFF caractères italiques (avec voyelles accentuées)

Une police de caractères est activée par la frappe de ESC, suivi du numéro (1-5) de la police.
Si l'on veut taper un seul caractère des polices 2 ou 3 on pourra utiliser :

CTRL G : caractère grec ou graphique

CTRL I ou bouton manette 0 ou ⌂ : caractère en indice

CTRL X ou bouton manette 1 ou ⌃ : caractère en exposant.



La table de formes

Pour afficher des symboles plus complexes, *EDIGRAPH* utilise aussi une table de formes classique. Elle est chargée à l'adresse $6000 et comporte provisoirement 2 formes chimiques (benzène et cyclohexane) et 6 formes d'électronique (résistance, interrupteur, générateur, etc.).
Une forme donnée est appelée par ESC suivi de F (ou G) et du numéro de la forme. F trace la forme horizontalement (ROT = 0), G la trace verticalement (ROT = 16).

Le traitement de blocs

EDIGRAPH permet le traitement de zones rectangulaires de l'écran graphique . Ces 'blocs' pourront être déplacés, dupliqués, effacés, inversés, sauvés sur disquette et réincorporés à d'autres pages graphiques.

Déplacer le curseur au coin supérieur gauche de la zone à traiter et taper ESC B. Le curseur se réduit à un point.

a) saisie du bloc dans le tampon : Déplacer le curseur au coin inférieur droit du bloc. Taper ESC W pour recopier le contenu de la zone ainsi délimitée dans le tampon. Donner un nom de fichier pour sauver le bloc sur disquette (le fichier s'appelera BLOC.nom), sinon taper simplement ⌟.

b) mettre une zone écran en inverse : le curseur ayant été amené au coin inférieur droit , taper ESC N.

c) effacer une zone : comme précédemment, mais taper ESC C.

d) déplacer un bloc : Taper ESC B après positionnement du curseur au coin inférieur droit. Déplacer le curseur jusqu'à la nouvelle position du bloc et taper ESC D. La zone initiale sera effacée.

e) copier un bloc : comme ci-dessus, mais taper ESC C à la nouvelle position. L'ancienne zone ne sera pas effacée. Une autre possibilité de duplication est de rappeler la zone tampon .

f) rappel d'un bloc : Ce traitement est le seul à ne pas être activé par ESC B. Amener le curseur à la position souhaitée (correspondant au coin supérieur gauche du bloc) et taper ESC R. La frappe de ⌟ copiera la zone tampon, sinon donner le nom du fichier à rappeler de la disquette.

Remarques :
La procédure de traitement des blocs est lente, donc patience. Aucun test n'est effectué pour savoir si le bloc recopié pourra se loger à sa nouvelle place.

Les commandes

EDIGRAPH utilise trois niveaux d'analyse des commandes :

1 : commandes par la touche CTRL , principalement pour le déplacement du curseur , et pour obtenir les caractères non disponibles au clavier du][+

2 : commandes activées par ESC : changement de police, changement du mode d'affichage (normal,double hauteur,inverse).

3 : enfin les touches F et G tapées après ESC, permettent d'afficher la forme dont le numéro suit.
Les touches Ó et ⌘ permettent respectivement l'écriture en indice ou en exposant.

Fonctions activées par la touche ESC

ESC 1	caractères français
ESC 2	caractères grecs et graphiques
ESC 3	indices ou exposants
ESC 4	caractères épais
ESC 5	caractères italiques
ESC B	activation du mode 'BLOC'
ESC C	copie d'un bloc à la nouvelle position du curseur
ESC D	deplacement d'un bloc à la nouvelle position du curseur
ESC E	effacement d'un bloc-écran
ESC N	négatif d'un bloc
ESC F	tracé d'une forme avec ROT=0. A faire suivre du numéro de la forme
ESC G	tracé d'une forme avec ROT=16. Faire suivre du numéro de la forme
ESC H	vide l'écran HGR et ramène le curseur au point supérieur gauche
ESC I	affiche les instructions.
ESC L	chargement d'une image graphique en page 2.
ESC P	hardcopy
ESC Q	sortie du programme.
ESC S	sauvegarde de la page graphique
ESC V	bascule vidéo normale/inverse
ESC Y	bascule hauteur normale/double.
ESC -	bascule soulignement
ESC /	affiche le symbole 'intégrale'

Fonctions activées par la touche contrôle

CTRL D	affiche le catalogue de la disquette.
CTRL G	le prochain caractère frappé sera pris dans la police 2 .
CTRL H	déplacement du curseur vers la gauche. Avantageusement remplacé par ⇐
CTRL I	le prochain caractère sera placé en indice (police 3A)
CTRL J	déplacement du curseur vers le bas . (utiliser ⇓ sur le //e)
CTRL K	déplacement du curseur vers le haut. (utiliser ⇑ sur le //e)
CTRL S	bascule majuscule/minuscule (utile seulement sur le][+)
CTRL X	le prochain caractère sera en exposant (police 3B)
CTRL ^	accent circonflexe. Après la frappe de CTRL ^, le curseur recule d'une position. Puis taper la voyelle désirée

Des caractères non disponibles sur le clavier du][+ (é è [ ] ù ç °) peuvent être obtenus à l'aide de Shift M, Ctrl Shift M, Ctrl N, Ctrl O, Ctrl Z. Faites l'essai en mode minuscule et majuscule.

Structure du programme

L 200-390	sous-programmes d'écriture de caractères en mode graphique, placés en début du programme pour accélérer l'exécution
L 400-530	sous-programmes de traitement des blocs
L 600-790	gestion de l'affichage
L 900-990	le mode d'écriture inverse, normal ou double hauteur est affichée dans le coin inférieur droit de l'écran
L 1000-1900	positionnement et affichage des formes
L 3000-3100	passage en mode HGR2 sans effacement
L 4000-4100	remise à zéro de certaines variables
L 5000-5200	sauvegarde de la page graphique 2 sur disquette
L 6000-6200	rappel d'une image graphique en page 2
L 7000-7080	sauvegarde d'un bloc
L 8000-8070	lecture d'un bloc
L 9000-9100	traitement des erreurs DOS
L 10000-11000	analyse des touches frappées. Touches CTRL

L 12000-14000	commandes activées par la touche ESC
L 15000-18000	appel du programme de hardcopy (à installer par l'utilisateur)
L 18000-19000	saisie du numéro de forme
L 20000-20210	affichage des commandes
L 24000-26030	initialisation . Chargement des tables (caractères et formes). Pour une hardcopy, rajouter ici le chargement du sous-programme nécessaire
ligne 30000	redémarrage à chaud, sans perte des variables, par un GOTO 30000

Occupation de la mémoire

$800	EDIGRAPH
$2FFF	HIMEM
$3000	police n°1
$3300	police n°2
$3600	police n°3
$3900	police n°4
$3C00	police n°5
$4000	page graphique
$6000	table de formes
$7000	tampon
$9000	fin du tampon

EDIGRAPH a intentionnellement été écrit en BASIC sans utiliser de sous-programmes assembleur, pour permettre à chacun de l'adapter à ses propres besoins. Il souffre donc de la lenteur de ce langage, particulièrement pour l'écriture en double hauteur et le traitement des blocs.

Pour obtenir malgré tout une vitesse d'exécution acceptable, tous les programmes de tracé ont été placés en tête du programme et optimisés au maximum. Le BASIC est suffisamment rapide pour permettre une vitesse moyenne de frappe des caractères en taille normale. En mode double hauteur, l'affichage est plus lent et pourrait bénéficier d'un traitement en assembleur. La lenteur d'exécution qui n'est pas un handicap lors de la phase d'apprentissage pourra etre supprimée si vous disposez d'un compilateur.

Remarques

Sur un Apple][+ les manettes doivent être connectées. Sinon, supprimer la ligne 10040.

EDIGRAPH a été développé sur un][+ et ses commandes adaptées à son clavier. Il utilise les possibilités supplémentaires du clavier du //e, mais l'utilisateur préfèrera peut-être modifier l'attribution de certaines touches.

Pour utiliser *EDIGRAPH*, faire RUN EDIGRAPH. Les fichiers SET et FORMES4 sont chargés par le programme.

En cas de 'plantage', taper CTRL RESET, puis GOTO 30000. L'image HGR en mémoire n'est pas altérée. Le curseur est repositionné dans le coin supérieur gauche de l'écran.

Programme Edigraph

```
40  HIMEM: 12287
100 REM  *************************
110 REM  *                       *
120 REM  *    EDIGRAPH - V.2     *
130 REM  *                       *
140 REM  *     R.JOST  - 1985    *
150 REM  *                       *
160 REM  *************************
170 GOSUB 24000: GOSUB 20010: GOSUB 3020
    :ZE = 49: GOSUB 910: GOTO 10000: END
200 :
210 REM  SOUS-PROGRAMMES DE TRACE DE LET
    TRES
220 :
230 FOR J = NK TO NJ: POKE BA + Z1(J), P
    EEK (ZX + J): NEXT : RETURN : REM
      MODE NORMAL
240 FOR J = NK TO NJ: POKE BA + Z1(J),Z1
    % -  PEEK (ZX + J): NEXT : RETURN :
    REM MODE INVERSE
250 ZV = ZV - 1:BA =  FN B1(ZH): FOR J =
     NK TO 7: POKE BA + Z1(J), PEEK (ZX
    +  INT (J / 2)): NEXT :ZV = ZV + 1:B
    A =  FN B1(ZH): FOR J = 8 TO NJ + 8:
     POKE BA + Z1 * (J - 8), PEEK (ZX +
     INT (J / 2)): NEXT : RETURN : REM
    AFFICHAGE DOUBLE HAUTEUR
260 ZV = ZV - 1:BA =  FN B1(ZH): FOR J =
     NK TO 7: POKE BA + Z1(J),Z1% -  PEE
    K (ZX +  INT (J / 2)): NEXT :ZV = ZV
     + 1:BA =  FN B1(ZH): FOR J = 8 TO N
    J + 8: POKE BA + Z1 * (J - 8),Z1% -
     PEEK (ZX +  INT (J / 2)): NEXT : RE
    TURN : REM  INVERSE ET DOUBLE HAUTEU
    R
270 BA =  FN B1(ZH)
280 CZ =  PEEK (BA + ZZ)
290 ZA = 12799 + 28 * (FL <  > 0): POKE
    BA + ZZ, PEEK (ZA): RETURN : REM AFF
    ICHAGE DU CURSEUR
300 POKE BA + ZZ,CZ: RETURN : REM  EFFAC
    EMENT DU CURSEUR
400 REM  TRAITEMENT DES BLOCS
410 DX = NX - X:DY = NY - Y + 1
420 GOSUB 300
430 ZW = ZV
440 FOR K = 0 TO DY::ZV = Y + K:BA =  FN
     B1(X):ZV = ZW + K:BB =  FN B1(ZH)
450 FOR I = 0 TO DX:BC = BA + I:BD = BB
    + I
460 FOR J = 0 TO 7:CH =  PEEK (BC + Z1(J
    ))
470 IF FL = 7 THEN TA = TA + 1: POKE TA,
    CH: GOTO 520
475 IF ZC = 67 THEN  POKE BD + Z1(J),CH:
     GOTO 520
480 IF ZC = 68 THEN  POKE BC + Z1(J),0:
```

```
    POKE BD + Z1(J),CH: GOTO 520
490 IF FL = 6 THEN TA = TA + 1: POKE BD
    + Z1(J), PEEK (TA): GOTO 520
500 IF ZC = 78 THEN  POKE BC + Z1(J),Z1%
     - CH: GOTO 520
510 IF ZC = 69 THEN  POKE BC + Z1(J),0:
    GOTO 520
520 NEXT : POKE  - 16336, PEEK ( - 16336
    ): NEXT : NEXT
530 ZV = ZV + 1:FL = 0: GOSUB 270:ZV = Z
    W: RETURN
600 :
610 REM  SOUS-PROGRAMME D'AFFICHAGE
620 :
630 Z =  PEEK ( - 16336) -  PEEK ( - 163
    36)
640 BA =  FN B1(ZH)
645 IF ZC = 30 THEN ZC = 62: GOTO 660
650 ZC = ZC - 32 + 32 * ((ZF = 1) AND (Z
    C > 64 AND ZC < 97))
660 IF F1 = 1 THEN WB = ZB:ZB = 13824
670 IF F1 = 2 THEN WB = ZB:ZB = 13826
680 IF F1 = 3 THEN WB = ZB:ZB = 13056
690 ZX = ZB + 8 * ZC
700 IF ZM = 1 AND ZL = 1 THEN  GOSUB 260
    : GOTO 740
710 IF ZM = 1 THEN  GOSUB 290: GOSUB 240
    : GOTO 740
720 IF ZL <  > 0 THEN  GOSUB 250: GOTO 7
    40
730 GOSUB 230
740 NK = 2 * (F1 = 10): IF F1 = 10 THEN
     GOSUB 300:ZH = ZH - 1: GOSUB 290:F1
     = 0
745 BA = BA + 1
750 ON ZH < 40 GOSUB 290
760 ZH = ZH + 1: IF ZH > 40 THEN ZH = 1:
    ZV = ZV + 1 + (ZL = 1): GOSUB 270
770 IF ZH = 37 THEN  PRINT  CHR$ (7);
780 IF F1 <  > 0 THEN ZB = WB:F1 = 0
790 RETURN
900 REM  AFFICHAGE DU MODE DANS LA CASE
    40,24
910 ZX = 12288 + 1536 * (ZL = 0) + 8 * (
    ZE - 32)
920 IF ZM = 1 THEN  FOR J = 0 TO NJ: POK
    E 17399 + Z1(J),Z1% -  PEEK (ZX + J)
    : NEXT : RETURN
930 FOR J = 0 TO NJ: POKE 17399 + Z1(J),
     PEEK (ZX + J): NEXT : RETURN
940 REM  EFFACEMENT DE LA CASE 40,24
950 BX = BA:BA = 17399: FOR J = 0 TO NJ:
     POKE BA + Z1(J), PEEK (ZB + J): NEX
    T :BA = BX: RETURN
960 END
1000 :
1010 REM *****  DESSIN D'UN CYCLE *****
1020 :
1030 IF SH > 2 THEN 1070
1040 GOSUB 300:XZ = ZH * 7 - 5 + 5 * (Z
     B = 13056 OR F1 = 1):YZ = ZV * 8 -
      16 * (F1 = 0)
1050 IF XZ < 1 OR YZ < 1 OR XZ > 256 OR
      YZ > 168 THEN  RETURN
1060 DRAW SH AT XZ,YZ:ZH = ZH + 4 * (F1
     = 0):F1 = 0: GOSUB 270: RETURN
1070 REM  ELECTRONIQUE
1080 GOSUB 300:XZ = ZH * 7 - 6 + 2 * (Z
     C = 71)
1090 YZ = ZV * 8 - 5 - 3 * (ZC = 71)
1100 DRAW SH AT XZ,YZ:ZH = ZH + 3 * (SH
      = 6 OR SH = 7 OR SH = 8) + 4 * (S
     H = 4) + 5 * (SH = 5): GOSUB 270:
     RETURN
3000 REM  ***** TEXT -> HGR2 *****
3010 :
3020 POKE 49232,0: POKE 49234,0: POKE 4
     9237,0: POKE 49239,0: GOSUB 900: R
     ETURN
4000 :
4010 :
4020 REM  ***** MISE A ZERO DES POINTEU
     RS *****
4030 ZE = 49:ZF = 0:ZM = 0:F1 = 0:ZL =
     0:ZB = 12288:TA = 28676:F2 = 0:F3
     = 0:ZH = 1:ZV = 1:FL = 0: RETURN
5000 REM
5010 REM *****  SAUVE PAGE GRAPHIQUE **
5020 :
5030 ONERR  GOTO 9000
5040 PRINT  CHR$ (7);: GOSUB 300: TEXT
     : HTAB 1: VTAB 23: CALL  - 868: IN
     PUT "SAUVEGARDE DE LA PAGE ";N$: I
     F N$ = "" THEN 5060
5050 GOSUB 300: PRINT D$"BSAVE "N$",A$4
     000,L$1FF8": GOSUB 3020
5060 GOSUB 3020
5070 RETURN
6000 :
6010 REM ***** CHARGE PAGE GRAPHIQUE **
6020 :
6030 ONERR  GOTO 9000
6040 PRINT  CHR$ (7);: TEXT : HTAB 1: V
     TAB 23: CALL  - 868: INPUT "CHARGE
     MENT DE LA PAGE ";N$: IF N$ = "" T
     HEN 6060
6050 PRINT D$"BLOAD "N$",A$4000"
6060 GOSUB 3020: GOSUB 290
6070 RETURN
7000 :
7010 REM ***** SAUVE UN BLOC *****
7020 :
7030 ONERR  GOTO 9000
7040 PRINT  CHR$ (7);: TEXT : HTAB 1: V
     TAB 23: CALL  - 868: INPUT "SAUVEG
     ARDE DU BLOC ";N$: IF N$ = "" THEN
      5060
7050 GOSUB 300
```

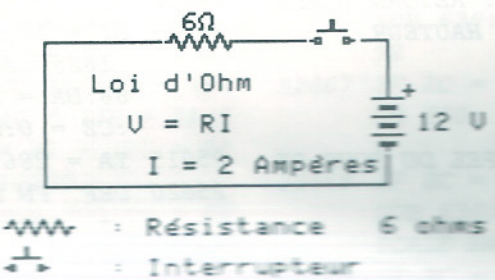
```
7060 PRINT D$"BSAVE "N$".BLOC,A$
     7000,L"(DX + 1) * (DY + 1) *
     8 + 97070  GOSUB 3020
7080 RETURN
8000 :
8010 REM ***** CHARGE UN BLOC *****
8020 :
8030 ONERR  GOTO 9000
8040 PRINT  CHR$ (7);: TEXT : HTAB 1: V
     TAB 23: CALL  - 868: INPUT "CHARGE
     MENT DU BLOC ";N$: IF N$ = "" THEN
      5060
8050 PRINT D$"BLOAD "N$".BLOC,A$7000"
8060 GOSUB 3020
8070 RETURN
9000 REM TRAITEMENT ERREURS
9010 TEXT : HOME
9020 ER =  PEEK (222):EL =  PEEK (218)
     + 256 *  PEEK (219)
9030 IF ER = 4 THEN  PRINT "DISK PROTEG
     E EN ECRITURE": GOTO 9080
9040 IF ER = 6 THEN  PRINT "CE FICHIER
     N'EXISTE PAS": GOTO 9080
9050 IF ER = 8 THEN  PRINT "ERREUR I/O"
     : GOTO 9080
9060 IF ER = 9 THEN  PRINT "DISK REMPLI
     ": GOTO 9080
9065 IF ER = 255 THEN  PRINT "TAPER UNE
      TOUCHE POUR CONTINUER": GOTO 9080
9070 PRINT "ERREUR "ER" DANS LA LIGNE "
     EL: END
9080 POKE 216,0: GET R$: HOME
9090 GOSUB 3020: GOTO 10030
10000 :
10010 REM  *** ANALYSE DES COMMANDES ***
10020 :
10030 WAIT  - 16384,128,127:ZC =  PEEK (
       - 16384) - 128: POKE  - 16368,0
10040 F1 = F1 + ( PEEK ( - 16287) > 127)
       + 2 * ( PEEK ( - 16286) > 127)
10045 IF ZC = 30 THEN F1 = 10: GOSUB 630
      : GOTO 10030
10050 IF ZC > 31 THEN  GOSUB 630: GOTO
       10030
10060 IF ZC = 4 THEN  PRINT : TEXT : HOM
      E : PRINT  CHR$ (4)"CATALOG": CALL
        - 651: GOSUB 3020: GOTO 10030
10070 IF ZC = 27 THEN  GOSUB 12010: GOTO
       10030
10080 :
10090 REM  DEPLACEMENT DU CURSEUR
10100 :
10110 IF ZC = 8 THEN  GOSUB 300:ZH = (ZH
       - 1) * (ZH > 1) + 40 * (ZH = 1):Z
      V = ZV - (ZH = 40) * (1 + (ZL = 1)
      ): GOSUB 270: GOTO 10030
10120 IF ZC = 10 THEN  GOSUB 300:ZV = ZV
       + (ZV < 24) + ((ZL = 1) AND (ZV <
       23)): GOSUB 270: GOTO 10030
10130 IF ZC = 11 THEN  GOSUB 300:ZV = ZV
       - (ZV > 1) - ((ZL = 1) AND (ZV >
      2)): GOSUB 270: GOTO 10030
10140 IF ZC = 13 THEN  GOSUB 300:ZH = 1:
      ZV = ZV + (ZV < 24) + ((ZL = 1) AN
      D (ZV < 23)): GOSUB 270: GOTO 1003
      0
10150 IF ZC = 21 THEN  GOSUB 300:ZH = (Z
      H + 1) * (ZH < 40) + (ZH = 40):ZV
      = ZV + (ZH = 1) * (1 + (ZL = 1)):
      GOSUB 270: GOTO 10030
10160 :
10170 REM  TOUCHES CTRL POUR APPLE II+
10180 :
10190 IF ZC = 9 THEN F1 = 1: GOTO 10030:
       REM CTRL-I => LETTRE SUIVANTE EN
      INDICE
10200 IF ZC = 14 THEN ZC = 95: GOSUB 630
10210 IF ZC = 15 THEN ZC = 92: GOSUB 630
10220 IF ZC = 16 THEN ZC = 96: GOSUB 630
10230 IF ZC = 19 THEN ZF = (ZF = 0): GOT
      O 10030: REM BASCULE MAJUSCULES/MI
      NUSCULES
10240 IF ZC = 24 THEN F1 = 2: GOTO 10030
10250 IF ZC = 7 THEN F1 = 3: GOTO 10030
10260 IF ZC = 26 THEN ZC = 126: GOSUB 63
      0
10270 IF ZC = 29 THEN ZC = 91: GOSUB 630
10280 GOTO 10030
12000 :
12010 REM  : LA TOUCHE ESC
12020 :
12030 WAIT  - 16384,128,127:ZC =  PEEK (
       - 16384) - 128: POKE  - 16368,0
12040 IF ZC < 49 OR ZC > 53 THEN 12060
12050 ZE = ZC: GOSUB 300:ZB = 12288 + 76
      8 * (ZC - 49): GOSUB 270:ZX = 1228
      8 + 8 * (ZC - 32): GOSUB 900: RETU
      RN
12060 IF ZC = 45 THEN NJ = 13 - NJ: RETU
      RN : REM TIRET -> SOULIGNE (BASCUL
      E)
12063 IF ZC = 43 THEN F1 = 3: GOSUB 630:
       RETURN : REM   + OU - PAR FRAPPE
      DE +
12065 IF ZC = 47 THEN F1 = 3:ZL = 1:ZC =
       64: GOSUB 630:ZL = 0: RETURN : RE
      M    SYMBOLE SOMME OBTENU PAR ESX
      /
12066 IF ZC = 21 THEN F1 = 3:ZC = 62: GO
      SUB 630: RETURN : REM  FLECHE DROI
      TE
12067 IF ZC = 8 THEN F1 = 3:ZC = 60: GOS
      UB 630: RETURN : REM  FLECHE GAUCH
      E
12068 IF ZC = 10 THEN F1 = 3:ZC = 59: GO
      SUB 630: RETURN : REM  FLECHE VERT
      ICALE BAS
12069 IF ZC = 11 THEN F1 = 3:ZC = 94: GO
```

```
SUB 630: RETURN : REM  FLECHE VERT
ICALE HAUT
12070 IF ZC = 66 THEN  GOSUB 300:FL = FL
 + 1:NX = ZH * (FL = 2):NY = (ZV -
 1) * (FL = 2): IF FL = 1 THEN X =
 ZH:Y = ZV + 1: GOSUB 270: RETURN
: REM : -> TRAVAIL SUR BLOCS
12075 IF FL = 3 THEN FL = 0
12080 IF ZC = 66 THEN  RETURN
12090 IF ZC = 70 THEN  ROT= 0: GOSUB 180
10: RETURN : REM F => FORME AVEC R
OT=0
12100 IF ZC = 71 THEN  ROT= 16: GOSUB 18
010: RETURN : REM G => FORME AVEC
ROT=16
12110 IF (ZC = 72 OR ZC = 104) THEN ZH =
 1:ZV = 1: HGR2 : GOSUB 270: RETUR
N : REM H(OME) => VIDE L'ECRAN HGR
12120 IF (ZC = 76 OR ZC = 108) THEN  GOS
UB 6010: RETURN
12130 IF (ZC = 81 OR ZC = 113) THEN  TEX
T : END : REM Q => QUITTER LE PROG
RAMME
12140 IF (ZC = 82 OR ZC = 114) THEN TA =
 28676: GOSUB 8030:DX =  PEEK (TA
- 1):DY =  PEEK (TA):X = ZH:Y = ZV
:FL = 6: GOSUB 420: RETURN : REM R
 => RAPPEL D'UN BLOC
12150 IF FL = 1 AND (ZC = 78 OR ZC = 69)
 THEN NX = ZH:NY = ZV - 1: GOSUB 4
10: REM  INVERSI
12160 IF FL = 2 AND (ZC = 67 OR ZC = 68)
 THEN  GOSUB 410: RETURN : REM COP
IE OU DEPLACEMENT BLOC
12170 IF (ZC = 83 OR ZC = 115) THEN  GOS
UB 5010: RETURN : REM S => SAUVE U
NE PAGE GRAPHIQUE
12180 IF (ZC = 73 OR ZC = 105) THEN  GOS
UB 20010: GOSUB 3020: RETURN : REM
 I(NSTRUCTIONS
12190 IF (ZC = 86 OR ZC = 118) THEN ZM =
 (ZM = 0): GOSUB 900: RETURN : REM
 BASCULE VIDEO INVERSE/NORMAL
12200 IF FL = 1 AND (ZC = 87 OR ZC = 119
) THEN TA = 28676:FL = 7:NX = ZH:N
Y = ZV - 1: GOSUB 410: POKE 28675,
DX: POKE 28676,DY: GOSUB 7030: RET
URN : REM W => SAUVEGARDE D'UN BLO
C
12210 IF (ZC = 89 OR ZC = 121) THEN ZL =
 (ZL = 0): GOSUB 900: RETURN : REM
 Y => BASCULE DOUBLE HAUTEUR
12220 RETURN
15000 :
15010 REM  HARD COPY
15020 REM  METTRE ICI L'APPEL DU SOUS-PR
OGRAMME DE HARDCOPY
15100 RETURN
18000 :
18010 REM  SHAPES
18020 :
18030 WAIT  - 16384,128,127:SH =  PEEK (
 - 16384) - 128 - 48: POKE  - 1636
8,0
18040 IF SH < 1 OR SH > 9 THEN 18030
18050 GOSUB 1020
18060 RETURN
20000 :
20010 REM  ***** INSTRUCTIONS *****
20020 :
20030 TEXT : HOME
20040 INVERSE : PRINT "EDITEUR HGR : EDI
GRAPH (C) 1985 R.JOST  ";: NORMAL
20050 INVERSE : PRINT "<ESC>:": NORMAL
20060 PRINT "1-5: POLICES 1 A 5. B: TRAI
TEMENT BLOCS"
20070 PRINT "F: FORMES 0 DEG.    G: FORM
ES 90 DEG"
20080 PRINT "H: HOME             I: INST
RUCTIONS"
20090 PRINT "L: CHARGE PAGE HGR. S: SAUV
E PAGE HGR"
20100 PRINT "R: LECTURE BLOC     P: HARD
COPY."
20110 PRINT "Q: QUITTER.         -: MODE
 SOULIGNE."
20120 PRINT "V: BASCULE VIDEO    Y: BASC
ULE HAUTEUR"
20130 PRINT : PRINT "SOUS-COMMANDES BLOC
S"
20140 PRINT "N: NEGATIF          D: DEPL
ACE UN BLOC."
20150 PRINT "C: COPIE BLOC       E: EFFA
CE UN BLOC."
20160 PRINT "W: ECRITURE BLOC."
20170 PRINT : INVERSE : PRINT "CTRL": NO
RMAL
20180 PRINT "S: BASCULE MAJ/MIN. G: LETT
RE GRECQUE."
20190 PRINT "I: INDICES.         X: EXPO
SANTS."
20200 PRINT "J: CURSEUR BAS.     K: CURS
EUR HAUT."
20210 PRINT "D: CATALOG"
20220 PRINT "<RETURN>: RETOUR CHARIOT"
20230 CALL  - 651
20240 RETURN
24000 :
24010 REM  ******  INITIALISATION  *****
24020 REM
25010 Z1 = 1024:Z1% = 255:Z2% = 128:Z2 =
 16255:Z3 = 984:ZZ = 7168:ZB = 122
88:BA = 16384:ZH = 1:ZV = 1:ZX = 0
:CZ = 0:NJ = 7:B1(0) = 0:ZM = 0:
25015 TA = 28676: REM  ZONE TAMPON
25020 DEF  FN B1(Z) = Z2 + Z2% * ZV - Z3
 *  INT ((ZV - 1) / 8) + Z
25030 Z1(0) = 0:Z1(1) = 1024:Z1(2) = 204
```

```
      8:Z1(3) = 3072:Z1(4) = 4096:Z1(5)
      = 5120:Z1(6) = 6144:Z1(7) = 7168
26000 D$ =  CHR$ (4)
26010 PRINT D$"BLOAD SET,A$3000"
26020 PRINT D$"BLOAD  FORMES4,A$6000": P
      OKE 232,0: POKE 233,96: SCALE= 1:
      HCOLOR= 3
26030 RETURN
30000 REM  REDEMARRAGE A CHAUD
30010 :
30020 HOME : GOSUB 4030: GOSUB 3020: GOT
      O 10030
```

Polices 'SET'

```
3000- 00 00 00 00 00 00 00 00
3008- 08 08 08 08 08 00 08 00
3010- 14 14 14 00 00 00 00 00
3018- 00 08 08 08 08 08 08 00
3020- 08 3C 0A 1C 28 1E 08 00
3028- 06 26 10 08 04 32 30 00
3030- 04 0A 0A 04 2A 12 2C 00
3038- 08 08 08 00 00 00 00 00
3040- 08 04 02 02 02 04 08 00
3048- 08 10 20 20 20 10 08 00
3050- 08 2A 1C 08 1C 2A 08 00
3058- 00 08 08 3E 08 08 00 00
3060- 00 00 00 00 08 08 04 00
3068- 00 00 00 3E 00 00 00 00
3070- 00 00 00 00 00 00 08 00
3078- 00 20 10 08 04 02 00 00
3080- 1C 22 32 2A 26 22 1C 00
3088- 08 0C 08 08 08 08 1C 00
3090- 1C 22 20 18 04 02 3E 00
3098- 3E 20 10 18 20 22 1C 00
30A0- 10 18 14 12 3E 10 10 00
30A8- 3E 02 1E 20 20 22 1C 00
30B0- 38 04 02 1E 22 22 1C 00
30B8- 3E 20 10 08 04 04 04 00
30C0- 1C 22 22 1C 22 22 1C 00
30C8- 1C 22 22 3C 20 10 0E 00
30D0- 00 00 08 00 08 00 00 00
30D8- 00 00 08 00 08 08 04 00
30E0- 10 08 04 02 04 08 10 00
30E8- 00 00 3E 00 3E 00 00 00
30F0- 04 08 10 20 10 08 04 00
30F8- 1C 22 10 08 08 00 08 00
3100- 10 20 1C 20 3C 22 3C 00
3108- 08 14 22 22 3E 22 22 00
3110- 1E 22 22 1E 22 22 1E 00
3118- 1C 22 02 02 02 22 1C 00
3120- 1E 22 22 22 22 22 1E 00
3128- 3E 02 02 1E 02 02 3E 00
3130- 3E 02 02 1E 02 02 02 00
3138- 3C 02 02 02 32 22 3C 00
3140- 22 22 22 3E 22 22 22 00
3148- 1C 08 08 08 08 08 1C 00
3150- 20 20 20 20 20 22 1C 00
3158- 22 12 0A 06 0A 12 22 00
3160- 02 02 02 02 02 02 3E 00
3168- 22 36 2A 2A 22 22 22 00
3170- 22 22 26 2A 32 22 22 00
3178- 1C 22 22 22 22 22 1C 00
3180- 1E 22 22 1E 02 02 02 00
3188- 1C 22 22 22 2A 12 2C 00
3190- 1E 22 22 1E 0A 12 22 00
3198- 1C 22 02 1C 20 22 1C 00
31A0- 3E 08 08 08 08 08 08 00
31A8- 22 22 22 22 22 22 1C 00
31B0- 22 22 22 22 22 14 08 00
31B8- 22 22 22 2A 2A 36 22 00
31C0- 22 22 14 08 14 22 22 00
31C8- 22 22 14 08 08 08 08 00
31D0- 3E 20 10 08 04 02 3E 00
31D8- 08 14 08 00 00 00 00 00
31E0- 00 3C 02 02 02 3C 10 08
31E8- 20 10 1C 22 3E 02 3C 00
31F0- 14 22 1C 22 3E 02 3C 00
31F8- 00 00 00 00 00 00 00 7F
3200- 00 02 04 08 10 20 00 00
3208- 00 00 1C 20 3C 22 3C 00
3210- 02 02 1E 22 22 22 1E 00
3218- 00 00 3C 02 02 02 3C 00
3220- 20 20 3C 22 22 22 3C 00
3228- 00 00 1C 22 3E 02 3C 00
3230- 18 24 04 1E 04 04 04 00
3238- 00 00 1C 22 22 3C 20 1C
3240- 02 02 1E 22 22 22 22 00
3248- 08 00 0C 08 08 08 1C 00
3250- 10 00 18 10 10 10 12 0C
3258- 02 02 22 12 0E 12 22 00
3260- 0C 08 08 08 08 08 1C 00
3268- 00 00 36 2A 2A 2A 22 00
3270- 00 00 1E 22 22 22 22 00
3278- 00 00 1C 22 22 22 1C 00
3280- 00 00 1E 22 22 1E 02 02
3288- 00 00 3C 22 22 3C 20 20
3290- 00 00 3A 06 02 02 02 00
3298- 00 00 3C 02 1C 20 1E 00
32A0- 04 04 1E 04 04 24 18 00
32A8- 00 00 22 22 22 32 2C 00
32B0- 00 00 22 22 22 14 08 00
32B8- 00 00 22 22 2A 2A 36 00
32C0- 00 00 22 14 08 14 22 00
32C8- 00 00 22 22 22 3C 20 1C
32D0- 00 00 3E 10 08 04 3E 00
32D8- 20 10 1C 22 3E 02 3C 00
32E0- 04 08 22 22 22 32 2C 00
32E8- 10 20 1C 22 3E 02 3C 00
32F0- 2C 1A 00 00 00 00 00 00
32F8- 7F 7F 7F 7F 7F 7F 7F FF
3300- 00 00 00 00 00 00 00 00
3308- 08 08 08 08 08 08 08 08
3310- 00 2A 2A 2A 2A 2A 2A 00
3318- 00 14 14 14 14 14 14 00
3320- 40 00 30 00 1C 00 1F 00
3328- 60 38 0E 07 03 01 00 00
3330- 00 00 36 49 49 36 00 00
3338- 00 00 00 00 00 00 00 00
3340- 00 20 10 08 04 02 00 00
3348- 00 02 04 08 10 20 00 00
3350- 1C 14 14 14 1C 00 00 00
3358- 08 08 3E 08 08 00 3E 00
3360- 00 04 08 12 24 08 10 00
3368- 00 00 00 7F 00 00 00 00
3370- 10 08 48 24 24 12 12 08
3378- 00 10 08 24 12 08 04 00
3380- 00 00 00 00 00 00 00 7F
3388- 40 20 10 08 04 02 01 00
3390- 7F 00 00 00 00 00 00 00
3398- 01 02 04 08 10 20 40 00
33A0- 01 01 01 01 01 01 01 01
33A8- 40 40 40 40 40 40 40 40
33B0- 00 00 00 78 08 08 08 08
33B8- 00 00 00 0F 08 08 08 08
33C0- 08 08 08 78 00 00 00 00
33C8- 08 08 08 0F 00 00 00 00
33D0- 08 08 08 08 08 2A 1C 08
33D8- 00 14 14 14 14 14 14 00
33E0- 00 04 02 7F 02 04 00 00
33E8- 10 20 7F 00 7F 02 04 00
33F0- 00 10 20 7F 20 10 00 00
33F8- 71 11 11 0A 0A 0A 04 04
3400- 60 10 10 08 08 04 04 03
3408- 08 8C 14 92 3E A1 63 00
3410- 1F 22 22 1E 22 22 1F 00
3418- 3F 22 02 02 02 02 83 00
3420- 08 8C 14 92 22 A1 7F 00
3428- 3F 22 02 0E 02 22 3F 00
3430- 1C 08 3E 6B 3E 08 1C 00
3438- 3F 22 02 02 02 02 83 00
3440- B3 22 22 3E 22 22 B3 00
3448- 8C 08 08 08 08 08 8C 00
3450- 1C 22 A1 BF A1 22 1C 00
3458- B3 12 0A 06 0A 12 B3 00
3460- 08 8C 14 92 22 A1 63 00
3468- B3 36 2A 2A 22 22 B3 00
3470- B3 22 26 2A 32 22 B3 00
3478- 1C 22 A1 A1 A1 22 1C 00
3480- BF 22 22 22 22 22 B3 00
3488- 3F 21 00 1E 00 21 3F 00
3490- 1F 22 22 1E 02 02 83 00
3498- 3F 22 04 08 04 22 3F 00
34A0- BF 08 08 08 08 08 8C 00
34A8- 36 AD 08 08 08 08 8C 00
34B0- 63 A1 22 92 14 8C 08 00
34B8- 1C 22 A1 A1 22 92 B3 00
34C0- B3 22 14 08 14 22 B3 00
34C8- 1C 08 6B 2A 1C 08 8C 00
34D0- BF 22 10 08 04 22 BF 00
34D8- 3E 06 06 06 06 06 3E 00
34E0- 00 02 04 08 10 20 00 00
34E8- 3E 30 30 30 30 30 3E 00
34F0- 08 1C 2A 08 08 08 08 08
34F8- 00 00 00 00 00 00 00 7F
3500- 00 04 08 12 24 08 10 00
3508- 00 00 00 2E 11 19 B7 00
3510- 00 B8 A4 3C A2 26 9D 01
3518- 00 00 A6 28 98 10 88 00
3520- 18 84 08 8E 12 12 0C 00
3528- 00 00 9C 82 1E 02 8E 00
3530- 90 10 3C AA 2A 9E 84 04
3538- 00 00 A6 28 98 10 88 00
3540- 00 00 36 A4 24 92 10 88
3548- 00 00 88 08 84 04 0C 00
3550- 30 28 24 9E 12 0A 06 00
3558- 00 00 24 8A 06 89 31 00
3560- 06 84 08 8C 14 92 22 00
3568- 00 00 A4 24 92 2A 81 01
3570- 00 00 A3 24 14 0C 04 00
3578- 00 00 8C 92 22 92 8C 00
3580- 00 00 BF 15 14 14 92 00
3588- 08 38 84 9C 04 82 1C 10
3590- 00 00 18 24 92 0E 81 01
3598- 00 00 00 BE 89 09 87 00
35A0- 00 00 BF 09 84 04 82 00
35A8- 00 00 A3 24 92 8A 86 00
35B0- 00 00 A3 24 14 0C 04 00
35B8- 00 00 00 23 29 2D 9B 00
35C0- 00 00 91 14 8C 8C 14 A2
35C8- 00 10 89 2A 95 8F 82 02
35D0- 04 9C 04 82 82 04 84 06
35D8- 38 0C 0C 06 0C 0C 38 00
35E0- 08 08 08 08 08 08 08 08
35E8- 0E 18 18 30 18 18 0E 00
35F0- 2C 1A 00 3E 00 00 00 00
35F8- 7F 7F 7F 7F 7F 7F 7F 7F
3600- 00 00 00 00 00 00 00 00
3608- 00 00 08 08 08 00 08 00
3610- 00 00 14 14 00 00 00 00
3618- 00 00 10 3C 08 1E 04 00
3620- 00 00 08 1C 08 1C 08 00
3628- 00 00 14 10 08 04 14 00
3630- 00 00 1C 14 1C 00 00 00
3638- 00 00 08 08 00 00 00 00
3640- 00 00 08 04 04 04 08 00
3648- 00 00 08 10 10 10 08 00
3650- 00 00 00 14 08 14 00 00
3658- 00 00 00 08 1C 08 00 00
3660- 00 00 00 00 10 10 08 00
3668- 00 00 00 00 1C 00 00 00
3670- 00 00 00 00 00 08 00 00
3678- 00 00 10 10 08 04 04 00
3680- 00 00 1C 14 14 14 1C 00
3688- 00 00 0C 08 08 08 1C 00
3690- 00 00 1C 10 1C 04 1C 00
3698- 00 00 1C 10 18 10 1C 00
36A0- 00 00 04 14 1C 10 10 00
36A8- 00 00 1C 04 1C 10 1C 00
36B0- 00 00 1C 04 1C 14 1C 00
36B8- 00 00 1C 10 08 08 08 00
36C0- 00 00 1C 14 1C 14 1C 00
36C8- 00 00 1C 14 1C 10 1C 00
36D0- 00 00 00 08 00 08 00 00
36D8- 00 00 00 00 08 00 08 04
36E0- 00 00 10 08 04 08 10 00
36E8- 00 00 00 1C 00 1C 00 00
36F0- 00 00 04 08 10 08 04 00
36F8- 00 00 1C 10 08 00 08 00
3700- 00 00 00 36 49 49 36 00
3708- 00 00 1C 14 1C 14 14 00
3710- 00 00 0C 14 0C 14 0C 00
3718- 00 00 1C 04 04 04 1C 00
3720- 00 00 0C 14 14 14 0C 00
3728- 00 00 1C 04 0C 04 1C 00
3730- 00 00 1C 04 0C 04 04 00
3738- 00 00 1C 04 04 1C 1C 00
3740- 00 00 14 14 1C 14 14 00
3748- 00 00 1C 08 08 08 1C 00
3750- 00 00 10 10 10 14 1C 00
3758- 00 00 14 0C 04 0C 14 00
3760- 00 00 04 04 04 04 1C 00
3768- 00 00 22 36 2A 22 22 00
3770- 00 00 12 16 1E 1A 12 00
3778- 00 00 1C 14 14 14 1C 00
3780- 00 00 1C 14 1C 04 04 00
3788- 00 00 1C 14 14 1C 1C 20
3790- 00 00 1C 14 0C 14 14 00
3798- 00 00 1C 04 1C 10 1C 00
37A0- 00 00 1C 08 08 08 08 00
37A8- 00 00 14 14 14 14 1C 00
37B0- 00 00 14 14 14 14 08 00
37B8- 00 00 22 22 2A 36 22 00
37C0- 00 00 14 14 08 14 14 00
37C8- 00 00 14 14 08 08 08 00
37D0- 00 00 1C 10 08 04 1C 00
37D8- 00 00 1C 04 04 04 1C 00
37E0- 00 00 04 04 08 10 10 00
37E8- 00 00 1C 10 10 10 1C 00
```



```
37F0- 00 00 08 1C 2A 08 08 00
37F8- 00 00 00 00 00 00 00 7F
3800- 20 20 20 20 20 20 20 00
3808- 00 00 00 1C 10 1C 1C 00
3810- 00 00 04 1C 14 14 1C 00
3818- 00 00 00 1C 04 04 1C 00
3820- 00 00 10 1C 14 14 1C 00
3828- 00 00 00 1C 1C 04 1C 00
3830- 00 00 18 08 0C 08 08 00
3838- 00 00 00 1C 14 1C 10 1C
3840- 00 00 04 1C 14 14 14 00
3848- 00 00 08 00 08 08 08 00
3850- 00 00 08 08 08 08 04 00
3858- 00 00 04 14 0C 0C 14 00
3860- 00 00 0C 08 08 08 1C 00
3868- 00 00 00 3E 2A 2A 2A 00
3870- 00 00 00 1C 14 14 14 00
3878- 00 00 00 1C 14 14 1C 00
3880- 00 00 00 1C 14 14 1C 04
3888- 00 00 00 1C 14 14 1C 10
3890- 00 00 00 1C 04 04 04 00
3898- 00 00 00 18 04 10 0C 00
38A0- 00 00 08 18 08 08 10 00
38A8- 00 00 00 14 14 14 1C 00
38B0- 00 00 00 14 14 14 08 00
38B8- 00 00 00 2A 2A 2A 14 00
38C0- 00 00 00 14 08 08 14 00
38C8- 00 00 00 14 14 1C 10 18
38D0- 00 00 00 1C 18 0C 1C 00
38D8- 00 00 00 00 00 00 00 00
38E0- 00 00 00 00 00 00 00 00
38E8- 00 00 00 00 00 00 00 00
38F0- 00 00 00 00 00 00 00 00
38F8- 00 3E 3E 3E 3E 3E 3E 00
3900- 00 00 00 00 00 00 00 00
3908- 1C 1C 1C 1C 00 1C 1C 00
3910- 36 36 24 36 00 00 00 00
3918- 12 3F 3F 12 3F 3F 12 00
3920- 0C 3F 03 3F 30 3F 0C 00
3928- 00 27 17 0F 3C 3A 39 00
3930- 06 09 02 04 2A 11 2E 00
3938- 0C 0C 08 0C 00 00 00 00
3940- 38 1C 0E 0E 0E 1C 38 00
3948- 0E 1C 38 38 38 1C 0E 00
3950- 08 2A 1C 3E 1C 2A 08 00
3958- 0C 0C 3F 3F 0C 0C 00 00
3960- 00 00 00 00 00 1C 18 0C
3968- 00 00 3E 3E 00 00 00 00
3970- 00 00 00 00 00 1C 1C 00
3978- 60 70 38 1C 0E 07 03 00
3980- 1E 33 33 33 33 33 1E 00
3988- 3C 36 33 30 30 30 30 00
3990- 1E 3F 33 38 0E 3F 3F 00
3998- 1E 3F 30 3E 30 3F 1E 00
39A0- 38 3C 36 33 3F 30 30 00
39A8- 3F 3F 03 1F 30 3F 1E 00
39B0- 1C 06 03 1F 33 33 1E 00
39B8- 3F 3F 30 18 0C 0C 0C 00
39C0- 1E 3F 33 1E 33 3F 1E 00
39C8- 1E 3F 33 3E 30 3E 1E 00
39D0- 00 1C 1C 00 00 1C 1C 00
39D8- 00 1C 1C 00 00 1C 18 0C
39E0- 70 38 1C 0E 1C 38 70 00
39E8- 00 3E 3E 00 3E 3E 00 00
39F0- 07 0E 1C 38 1C 0E 07 00
39F8- 1E 3F 33 18 0C 00 0C 00
3A00- 38 40 1E 30 3E 33 3E 00
3A08- 1E 3F 33 3F 3F 33 33 00
3A10- 1F 3F 33 1F 33 3F 1F 00
3A18- 1E 3F 33 03 33 3F 1E 00
3A20- 1F 3F 33 33 33 3F 1F 00
3A28- 3F 3F 03 1F 03 3F 3F 00
3A30- 3F 3F 03 1F 1F 03 03 00
3A38- 1E 3F 03 3B 33 3F 1E 00
3A40- 33 33 33 3F 3F 33 33 00
3A48- 3F 3F 0C 0C 0C 3F 3F 00
3A50- B0 B0 B0 B0 B3 BF 9E 80
3A58- 33 3B 1F 0F 1F 3B 33 00
3A60- 03 03 03 03 03 3F 3F 00
3A68- 63 77 6B 63 63 63 63 00
3A70- 33 33 37 3F 3B 33 33 00
3A78- 1E 3F 33 33 33 3F 1E 00
3A80- 1F 3F 33 3F 1F 03 03 00
3A88- 1E 3F 23 23 2B 13 2E 00
3A90- 1F 3F 33 3F 1F 3B 33 00
3A98- 1E 3F 03 1E 30 3F 1E 00
3AA0- 3F 3F 0C 0C 0C 0C 0C 00
3AA8- 33 33 33 33 33 3F 1E 00
3AB0- 33 33 33 33 33 1E 0C 00
3AB8- 63 63 63 6B 6B 77 63 00
3AC0- 33 33 1E 0C 1E 33 33 00
3AC8- 33 33 33 1E 0C 0C 0C 00
3AD0- 3F 3F 18 0C 06 3F 3F 00
3AD8- 1F 1F 03 03 03 1F 1F 00
3AE0- 03 07 0E 1C 38 70 60 00
3AE8- 1F 1F 18 18 18 1F 1F 00
3AF0- 0C 1E 3F 0C 0C 0C 0C 00
3AF8- 00 00 00 00 00 00 00 7F
3B00- 06 0C 18 30 00 00 00 00
3B08- 00 00 1E 30 3E 33 3E 00
3B10- 03 03 1F 33 33 33 1F 00
3B18- 00 00 1E 33 03 33 1E 00
3B20- 30 30 3E 33 33 33 3E 00
3B28- 00 00 1E 33 1F 03 1E 00
3B30- 1C 36 06 1F 06 06 06 00
3B38- 00 00 1E 33 33 3E 30 1E
3B40- 03 03 1F 33 33 33 33 00
3B48- 00 0C 00 0C 0C 0C 0C 00
3B50- 30 00 30 30 30 33 33 1E
3B58- 03 03 33 1B 0F 1B 33 00
3B60- 0E 0C 0C 0C 0C 0C 1E 00
3B68- 00 00 3F 5B 5B 5B 5B 00
3B70- 00 00 1F 33 33 33 33 00
3B78- 00 00 1E 33 33 33 1E 00
3B80- 00 00 1F 33 33 1F 03 03
3B88- 00 00 3E 33 33 3E 30 30
3B90- 00 00 1F 33 03 03 03 00
3B98- 00 00 1E 03 1E 30 1E 00
3BA0- 06 06 1F 06 06 36 1C 00
3BA8- 00 00 33 33 33 33 3E 00
3BB0- 00 00 33 33 33 1E 0C 00
3BB8- 00 00 6D 6D 6D 6D 7E 00
3BC0- 00 00 33 1E 0C 1E 33 00
3BC8- 00 00 33 33 33 3E 30 1E
3BD0- 00 00 3F 18 0C 06 3F 00
3BD8- 0E 01 1E 33 1F 03 1E 00
3BE0- 38 00 33 33 33 33 3E 00
3BE8- 38 40 1E 33 1F 03 1E 00
3BF0- 00 00 28 14 00 00 00 00
3BF8- FF FF FF FF FF FF FF FF
3C00- 80 80 80 80 80 80 80 80
3C08- 90 98 18 8C 0C 80 06 00
3C10- B6 92 00 80 00 80 00 00
3C18- 80 A8 FE 94 BF 8A 00 00
3C20- 90 BC 02 9E 30 9F 04 00
3C28- C7 A5 97 88 F4 D2 F1 00
3C30- 8C 92 82 84 8A 92 AE 00
3C38- 98 88 00 80 00 80 00 00
3C40- E0 98 0C 86 06 86 0C 00
3C48- 98 B0 60 B0 30 8C 06 00
3C50- B0 D4 28 EB 14 95 06 00
3C58- 80 98 18 BF 0C 86 00 00
3C60- 80 80 00 80 00 8C 08 04
3C68- 80 80 00 BE 00 80 00 00
3C70- 80 80 00 80 00 80 0C 00
3C78- C0 A0 90 88 84 82 81 00
3C80- B8 E4 64 B2 32 99 0E 00
3C88- B0 B8 30 98 18 8C 0C 00
3C90- B8 E4 60 B8 0C 83 1F 00
3C98- B8 E4 60 9C 30 99 0E 00
3CA0- E4 B2 32 BF 18 8C 0C 00
3CA8- FC 84 9E B0 30 99 0E 00
3CB0- B8 E4 04 9E 32 99 0E 00
3CB8- FC E0 B0 98 0C 86 06 00
3CC0- B8 E4 64 BE 32 99 0E 00
3CC8- B8 E4 64 BC 30 98 0E 00
3CD0- 80 80 00 98 00 8C 00 00
3CD8- 80 80 00 98 00 8C 08 04
3CE0- B0 98 8C 86 8C 98 B0 00
3CE8- 80 80 3E 80 1F 80 00 00
3CF0- 86 8C 98 B0 98 8C 86 00
3CF8- 9C B2 30 88 08 80 0C 00
3D00- 30 40 38 20 3C 13 1E 00
3D08- B8 E4 64 BE 32 99 19 00
3D10- B8 E4 64 9E 32 B9 1F 00
3D18- F8 8C 04 82 02 83 1E 00
3D20- B8 E4 64 B2 32 99 0F 00
3D28- F8 8C 0C 9E 06 83 1F 00
3D30- F8 8C 0C 9E 06 83 03 00
3D38- B8 E4 04 82 3A 99 1E 00
3D40- CC E4 64 BE 32 99 19 00
3D48- B8 98 18 8C 0C 86 87 00
3D50- F0 E0 60 B0 32 99 0E 00
3D58- CC E4 34 8E 1A 99 31 00
3D60- 9C 8C 0C 86 06 83 1F 00
3D68- FC D4 54 EA 6A B1 31 00
3D70- CC EC 6C AA 3A 99 19 00
3D78- B8 E4 64 B2 32 99 0E 00
3D80- B8 E4 64 9E 02 81 01 00
3D88- B8 E4 64 B2 2A 99 2E 00
3D90- B8 E4 64 9E 0A 89 11 00
3D98- B8 E4 04 9C 30 99 0E 00
3DA0- FE 88 08 84 04 82 02 00
3DA8- CC E4 64 B2 32 99 0E 00
3DB0- C2 E2 B2 9A 8E 86 82 00
3DB8- C6 E2 6A B5 35 95 1E 00
3DC0- C6 E6 2C 9C 1C 9A 33 00
3DC8- C6 E4 B4 9C 8C 86 83 00
3DD0- FE B0 98 8C 86 83 3F 00
3DD8- F8 8C 0C 86 06 83 0F 00
3DE0- 01 02 04 08 10 20 40 00
3DE8- F8 E0 60 B0 30 98 1E 00
3DF0- B0 B8 54 88 08 84 04 00
3DF8- 00 00 00 00 00 00 00 7F
3E00- 00 00 00 00 00 00 00 00
3E08- 80 80 38 A0 3C 93 1E 00
3E10- 8C 84 1C 92 32 99 0F 00
3E18- 80 80 BC 86 83 83 1E 00
3E20- C0 E0 78 A6 26 93 1E 00
3E28- 80 80 9C B2 3E 83 1E 00
3E30- B8 CC 0C 9F 06 83 03 00
3E38- 80 80 38 A6 26 9E 10 0E
3E40- 8C 84 3C B2 32 99 19 00
3E48- 98 80 18 8C 0C 86 06 00
3E50- E0 80 60 B0 30 98 19 0E
3E58- 8C 84 64 9A 0E 8D 19 00
3E60- 90 98 18 8C 0C 86 06 00
3E68- 80 80 3E AA 2A 95 15 00
3E70- 80 80 3E B2 32 99 19 00
3E78- 80 80 1C 93 33 99 0E 00
3E80- 80 80 3C B2 32 8F 03 01
3E88- 80 80 38 A6 26 9E 18 B8
3E90- 80 80 78 8E 06 83 03 00
3E98- 80 80 9C A6 8C 99 8E 00
3EA0- 80 8C 3E 86 06 93 0E 00
3EA8- 80 80 4C A6 26 93 1E 00
3EB0- 80 80 B2 9A 8E 86 82 00
3EB8- 80 80 54 AA 2A 95 1E 00
3EC0- 80 80 22 9A 0C 8B 11 00
3EC8- 80 80 26 93 13 8F 08 07
3ED0- 80 80 3E 98 0C 83 1F 00
3ED8- 0C 02 1C 32 3E 03 1E 00
3EE0- 18 20 4C 26 26 13 1E 00
3EE8- 30 40 1C 32 3E 03 1E 00
3EF0- 02 06 0E 1E 0E 06 02 00
3EF8- 7F 7F 7F 7F 7F 7F 7F FF
```

Table 'FORMES4'

```
6000- 08 DB 12 00 E0 00 AC 01
6008- EE 01 15 02 5D 02 76 02
6010- 8D 02 49 49 2D 2D 2D 2D
6018- 2D 6D 49 D1 DB FB DB DB
6020- DB 1B DF 9B 49 09 4D 49
6028- 49 49 69 49 DA 1B DF DB
6030- DB DB DB DF 53 49 4D 49
6038- 2D 6D 49 69 89 DB DF 1B
6040- FF DB FF DB DF 4A 69 09
6048- 4D 49 49 4D 69 D1 FB 1B
6050- DF DB DB 1B DF FB 4A 4D
6058- 4D 49 49 49 69 69 1A DF
6060- FB DB DB DB 1B DF BB 69
6068- 69 49 49 49 49 69 A9 DF
6070- FB DB DB DB DB FB 1B 0E
6078- 4D 4D 49 49 49 49 4D D5
6080- DF FB DB DB DB 1B DF BB
6088- 09 4D 4D 49 49 49 69 69
6090- DA DF FB DB DB DB DF FB
6098- 4A 69 09 4D 49 49 4D 69
60A0- D1 1B DF 1B FF DB FF DB
60A8- DF 4A 09 4D 49 2D 6D 49
60B0- 69 89 DB FB DB DB DB DB
60B8- FB 9B 49 09 4D 49 49 49
60C0- 69 49 DA DB DF DB DB DB
60C8- FB DB 4A 49 29 2D 2D 2D
60D0- 2D 2D 4D 49 DA DB DB DB
60D8- DB DB DB DB 02 00 00 00
60E0- 49 49 2D 2D 2D 2D 2D 6D
60E8- 49 D1 DB FB DB DB DB 1B
60F0- DF 9B 49 09 4D 49 49 49
60F8- 69 49 DA 1B DF DB DB DB
6100- DB DF 53 49 4D 49 49 49
6108- 49 4D D1 1B DF DB DB DB
6110- DB 1B DF 4A 69 49 49 49
6118- 49 49 4D DA DF DB DB DB
6120- DB DB FB 4A 4D 49 49 49
6128- 49 49 69 1A DF DB DB DB
6130- DB DB DB 57 4D 49 49 49
6138- 49 49 49 15 DF DB DB DB
6140- DB DB DB 1B 0E 4D 49 49
6148- 49 49 49 49 D5 DF DB DB
6150- DB DB DB DB 57 69 49 49
6158- 49 49 49 09 8D 1B DF DB
6160- DB DB DB DB FB 4A 69 49
6168- 49 49 49 49 4D DA FB DB
6170- DB DB DB DB DF 4A 09 4D
6178- 49 49 49 49 4D D1 DB DF
6180- DB DB DB DB DF 53 49 69
6188- 49 49 49 09 4D 89 DB 1B
6190- DF DB DB DB DB FB DB 4A 49
6198- 29 2D 2D 2D 2D 2D 4D 49
61A0- DA DB DB DB DB DB DB DB
61A8- 02 00 00 00 24 2C 24 25
61B0- 25 2D 2C 2D 2D 2D 35 2D
61B8- 2E 2E 36 35 36 36 36 3E
61C0- 36 37 37 3F 3E 3F 3F 3F
61C8- 27 3F 3C 3C 24 27 24 2C
61D0- 2D 2D 2D 25 24 24 05 08
61D8- 08 09 39 3A 3A 3A 3A 32
61E0- 1A 32 36 36 09 2D 25 27
61E8- 2B 21 1F 23 3B 00 2D 24
61F0- 25 0E 36 35 2E 22 25 2C
61F8- 24 25 0E 36 35 2E 22 25
6200- 2C 24 25 0E 36 35 2E 22
6208- 25 2C 24 25 0E 36 35 2E
6210- 22 25 2C 2D 00 2D 24 25
6218- 0E 36 35 2E 22 25 2C 24
6220- 25 0E 36 35 2E 36 0A 3E
6228- 36 37 0D 09 3C 24 17 36
6230- 36 2E 2D 2D 2D 2D 2D 2D
6238- 2D 25 24 24 24 24 24 0C
6240- 09 3F 22 3C 36 23 34 3B
6248- 37 3E 26 23 3C 24 27 1E
6250- 36 37 3E 22 27 3C 24 27
6258- 1E 36 37 3E 00 2D 2E 25
6260- 3C 27 08 28 2D 2D 24 24
6268- 35 36 2E 2D 2D 1E 12 36
6270- 2D 24 37 09 2D 00 2D 2C
6278- 35 3E 27 01 08 08 2D 2C
6280- 25 2D 2C 35 11 12 36 2D
6288- 24 37 09 2D 00 2D 12 12
6290- 12 24 24 24 24 24 24 1F
6298- 23 3C 2F 01 20 0E 2A 0D
62A0- 09 12 12 32 36 36 2E 24
62A8- 24 24 05 08 08 30 36 36
62B0- 36 36 12 22 24 0D 21 24
62B8- 24 2C 36 36 36 16 0A 09
62C0- 24 24 24 24 24 24 0D 11
62C8- 12 36 36 36 25 24 24 2C
62D0- 12 2A 05 00 EE
```

Courrier automatisé

Véronique Schmitt

Lorsque nous abordons W.P.L. et le courrier dit "automatisé", nous ne trouvons que des "circulaires" à personnaliser et leur fichier d'adresses pré-enregistré. Or, si la circulaire répond bien aux besoins professionnels, pour lesquels la création d'un fichier est rentabilisée par des emplois successifs, elle paraît inadaptée pour des besoins personnels car, dans ce cas, le fichier n'est créé que pour la circulaire en question et tout le travail de saisie, qui est si fastidieux, n'est guère fructueux. En effet, dans le cas d'un particulier, de quelles "lettres-type" s'agit-il le plus souvent ?

- une lettre de demande de renseignements auprès d'agences de voyage;
- une demande de tarif à une compagnie d'assurance;
- une invitation des amis aux dix ans de la petite...
- etc.

Dans chaque cas, la liste des destinataires est différente et la mémoriser en vue d'une utilisation ultérieure s'avère souvent inutile (si ce n'est dans le dernier cas : le fichier "Amis" peut éventuellement resservir...).

Nous avons donc mis au point un programme W.P.L. de "courrier automatisé" pour satisfaire les besoins des particuliers.

Utilisation du programme

Comme pour tout programme W.P.L., il est nécessaire de charger préalablement Apple Writer IIe (en français).
Pour appeler le programme, tapez <P> (CONTROLE P), puis après apparition du message

Fichier de Glossaire 'ENTETE'

```
VVéronique SCHMITT
MMonsieur
CCher Monsieur
VVeuillez agréer,
```

Programme WPL 'FORMAT'

```
PND
PPR (ici mettre CTRL L, entre deux CTRL V bien sûr)
PPR                    ************************************************
PPR                    ***                                          ***
PPR                    ***       CHOIX DU FORMAT D'IMPRESSION       ***
PPR                    ***                                          ***
PPR                    ************************************************
PPR
PPR
PPR                     Avant de charger le format d'impression désiré,
PPR                    Vous etes-vous bien assuré de l'avoir sauvegardé
PPR                              sur la disquette W.P.L.
PPR
PPR
PIN NOM DU FORMAT D'IMPRESSION A CHARGER: =$A
PCS/$A//
PDO MENU
QC$A
PDO MENU
```

Programme WPL 'GLOSSAIRE'

```
PND
PPR (ici mettre CTRL L, entre deux CTRL V bien sûr)
PPR                    ************************************************
PPR                    ***                                          ***
PPR                    ***             CHOIX DU GLOSSAIRE           ***
PPR                    ***                                          ***
PPR                    ************************************************
PPR
PPR
PPR                          Avant de charger le glossaire désiré,
PPR                    Vous etes-vous bien assuré de l'avoir sauvegardé
PPR                              sur la disquette W.P.L.
PPR
PPR
PIN NOM DU GLOSSAIRE A CHARGER: =$A
```

IMPRESSION/PROGRAMME : DO MENU.

NDLR : Si, au lieu de baptiser le programme "MENU", nous l'appelons "STARTUP", Apple Writer le chargera automatiquement, lors du Boot.

Le programme se décompose en 4 phases principales :

- 1ère phase : chargement du format d'impression;
- 2ème phase : chargement du glossaire;
- 3ème phase : création de la "lettre-type" selon un schéma prédéterminé;
- 4ème phase : exploitation de la lettre créée (saisie des mentions variables et édition).

Il suffit ensuite de suivre pas à pas les instructions figurant à l'écran.

Fonctionnement du programme

Le menu

C'est un programme W.P.L. ne présentant aucune particularité. Pour quitter, il suffit de taper sur "RETURN".

Chargement du format d'impression et du glossaire

Ce sont deux programmes W.P.L. Il suffit d'entrer le nom du fichier. En cas d'erreur, un retour chariot permet de revenir au menu principal. Il est rappelé à l'utilisateur que les fichiers de format d'impression et les glossaires doivent être préalablement enregistrés sur la disquette. Les initiés à AppleWriter connaissent bien cette procédure de sauvegarde.

Création de la lettre

C'est un fichier texte. Il a été créé à partir d'AppleWriter IIe et se met donc à jour en utilisant les commandes AppleWriter.

A ce stade, l'utilisateur complète la lettre avec les mentions constantes (Réf., objet, date,

```
PCS/$A//
PDO MENU
QE $A
PDO MENU
```

Fichier AppleWriter 'LETTRETYPE'

```
Véronique SCHMITT
rue des Cerisiers
97212 AILLEURS

                                        (NOM)
                                        (ADRESSE)
                                        (VILLE)

Réf. :

Objet :

                                        Ailleurs,
                                        le

(APPELLATION)

Veuillez agréer, (APPELLATION), nos salutations distinguées.

                                        V. Schmitt,

1ère Phase : CREATION DE LA LETTRE TYPE
       - Taper <b> pour avoir le debut de la lettre
       - Compléter la lettre (Réf., Objet, Date, Corps)

2ème Phase : SAUVEGARDE DE LA LETTRE TYPE
       - Mettre le curseur au debut de la lettre type
       - Charger la lettre type completee sous le nom   XX
       - Taper <S> : XX/V. Schmitt,/

3ème Phase : RETOUR AU MENU WPL POUR EXPLOITER LA LETTRE TYPE
      - Taper <P> DO MENU
```

Programme WPL 'MENU'

```
PND
PPR (ici mettre CTRL L, entre deux CTRL V bien sûr)
PPR                 ************************************************
PPR                 ***                                          ***
PPR                 ***       COURRIER AUTOMATISE (W.P.L.)       ***
PPR                 ***                                          ***
PPR                 ************************************************
PPR
PPR                                   M E N U
PPR
PPR
PPR1- CHOIX DU FORMAT D'IMPRESSION
PPR
PPR2- CHOIX DU GLOSSAIRE
PPR
PPR3- CREATION D'UNE LETTRE TYPE
```

corps). Puis il la sauvegarde sous le nom de son choix (ici XX). Afin de ne pas annexer à la lettre créée les instructions qui figurent en-dessous, il ne faut sauvegarder le texte que de l'en-tête jusqu'au signataire, d'où la nécessité de placer le curseur au début de la lettre et de la sauvegarder sous le nom XX/V. Schmitt,/.

Enfin, comme il s'agit d'un fichier texte, il est nécessaire pour poursuivre de retaper <P> : DO MENU.

Remarque : Ce schéma de lettre concerne la correspondance de V. Schmitt. Le lecteur le personnalisera en tapant ses propres coordonnées à la place des siennes et sauvera le texte sous le nom "LETTRETYPE".

Exploitation de la lettre

C'est un programme W.P.L. qui se déroule en deux phases principales:

- saisie du nom de la lettre à éditer;
- saisie des mentions variables concernant le 1er destinataire, puis édition correspondante. Saisie des mentions variables concernant le destinataire suivant, puis édition, etc.

Le retour au menu se fait par un retour chariot lors de la saisie d'un nouveau nom de destinataire.

```
        PPR
        PPR4- EXPLOITATION D'UNE LETTRE TYPE
        PPR
        PPR
CHOIX   PIN                                     VOTRE CHOIX: =$A
        PCS/$A/1/
        PDO FORMAT
        PCS/$a/2/
        PDO GLOSSAIRE
        PCS/$a/3/
        PGO LT
        PCS/$a/4/
        PDO SAISIE
FIN     NO
        PQT
LT      NO
        L LETTRETYPE
```

Programme WPL 'SAISIE'

```
        PND
        PPR (ici mettre CTRL L, entre deux CTRL V bien sûr)
        PPR ************************************************************
        PPR ***                                                      ****
        PPR ***              EXPLOITATION D'UNE LETTRE TYPE          ****
        PPR ***                                                      ****
        PPR ************************************************************
        PPR
        PPR
        PIN NOM DE VOTRE LETTRE TYPE :=$c
BOUCLE  NO
        L $c
        PPR (ici mettre CTRL L, entre deux CTRL V bien sûr)
        B
        PIN NOM DU DESTINATAIRE :=$A
        PCS/$A//
        PDO MENU
        F/(NOM)/$A/a/
        B
        PIN ADRESSE :=$A
        F/(ADRESSE)/$A/a/
        B
        PIN VILLE :=$A
        F/(VILLE)/$A/a/
        B
        PIN APPELLATION :=$A
        F/(APPELLATION)/$A/a/
        B
        PNP
        PGO BOUCLE
```


Editeur Plein Ecran

EPE

Le Pacha

Apple][+, //e, //c

Listez vos programmes Basic en avant et en arrière.
Modifiez, insérez, effacez des caractères en plein écran sans relire les lignes.
Recherchez toute chaîne de caractères.
Choisissez vous-même les codes de contrôle d'EPE.
Modifiez EPE : le fichier source est sur la disquette.

150,00 F TTC franco (bon de commande page 74).

RWTS désassemblée

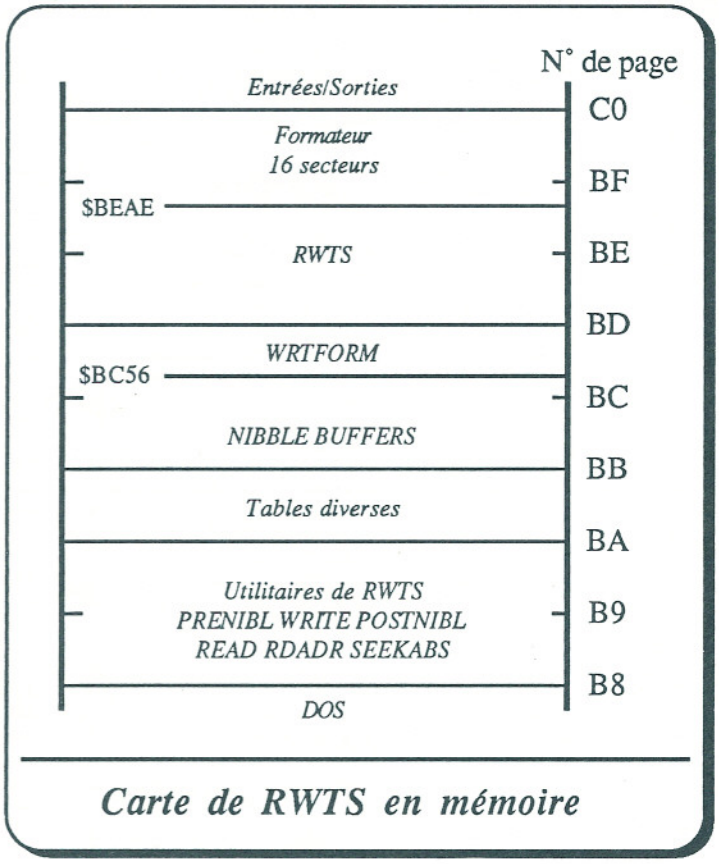
Patrice Neveu

Dans le numéro 9 de Pom's, Thierry Le Tallec désassemblait pour nous la ROM en $C600 : les méandres du Boot devenaient clairs... Aujourd'hui nous nous consacrons à Rwts. Beaucoup d'encre a coulé à son sujet, citons particulièrement celle de *Call Apple* dans *All About Dos*.

Les fichiers de désassemblage qui suivent sont écrits en Big Mac :

- Rwts 1 regroupe les sous-routines Prenibl Write Postnibl Read Rdadr Seekabs Wrtform.
- Rwts 2 est à proprement parler Rwts.
- Rwts 3 est le formatteur 16 secteurs du DOS 3.3.

Ces fichiers devraient permettre au lecteur de modifier avec plus de facilité le DOS. Ils devraient simplifier la mise au point de formattages fantaisistes, de protection logicielle...



Carte de RWTS en mémoire

RWTS 1
(Assembleur BigMac)

```
*********************************
*                               *
*            DISK II            *
* SOUS-ROUTINES UTILITAIRES DE  *
* READ / WRITE TRACK & SECTOR   *
*          ($B800.BCDE)         *
*                               *
*                               *
*     © PAR APPLE COMPUTER.     *
*                               *
*   DESASSEMBLE PAR P. NEVEU    *
*                               *
*********************************


            ORG $B800
            OBJ $8800

*==============================
*
*  LABELS
*
*==============================

BUFPTR    EQU $3E        POINTE SUR LA PAGE DE BUFFER DE
                         TRAVAIL
VOL       EQU $41        NUMERO DE VOLUME
SECTNO    EQU $3F        NOSECT
TRKNO     EQU $44        NO PISTE
TRKN      EQU $2A        PISTE DE DESTINATION
MONTIME   EQU $46        COMPTEUR SERVANT AUX DELAIS DU
                         MOTEUR
TEMP      EQU $2C
L27       EQU $27
COMPTEUR  EQU $26        COMPTEUR POUR LE BUFFER
                         SECONDAIRE
SLOTNUM   EQU $2B        NUMERO DE SLOT
CURTRK    EQU $0478      LE NUMERO DE LA PISTE ACTUELLE
L0678     EQU $0678
DATA1     EQU $BA11      BA11.BA28 : DELAIS POUR LES
                         MOUVEMENTS
DATA2     EQU $BA1D      DU BRAS. UTILISE LORS DES
                         PHASES ON/OFF
DATA3     EQU $BA29      TABLE DE CONVERSION AL'ECRITURE
NIBLBUF   EQU $BB00      ADRESSE DE DEBUT DU BUFFER
                         PRIMAIRE
NIBLBUF1  EQU $BBFF
NIBLBUF2  EQU $BC0       ADRESSE DE DEBUT DU BUFFER
                         SECONDAIRE
RDLTCH    EQU $C08C      SWITCH DE LECTURE DATA
WRLTCH    EQU $C08D      SWITCH D'ECRITURE DATA
RDMODE    EQU $C08E      SWITCH POUR MODE LECTURE
WRMODE    EQU $C08F      ECRITURE
PHSOFF    EQU $C080      ZONE ALLUMANT OU ETEIGNANT LES
```

```
                                4 PHASES
50  *                           ( $C080.C087)
51
52
53  *================================
54  * PRENIBL - WRITE - POSTNIBL
55  * READ    - RDADR - SEEKABS
56  *================================
57
58  *================================
59  * PRENIBL:NIBBLELIZE POUR ECRIRE
60  *         ENSUITE.
61  *================================
62
63  PRENIBL   LDX #$00          CONVERTIT 256 OCTETS (8 BITS)
64            LDY #$02          EN 342 NIBBLES (6 BITS)
65  PRENIB1   DEY               DE LA FORME 00XXXXXX
66            LDA (BUFPTR),Y    PAGE A CONVERTIR EST EN BUFPTR
67            LSR               METHODE DE CONVERSION 6+2 (DOS
                                3.3)
68            ROL NIBLBUF2,X    BUFFER SECONDAIRE
69            LSR
70            ROL NIBLBUF2,X
71            STA NIBLBUF,Y     BUFFER PRIMAIRE
72            INX
73            CPX #$56          CODE 85 OCTETS...
74            BCC PRENIB1
75            LDX #$00
76            TYA               ... 3 FOIS DE SUITE, POUR EN
                                CODER 256
77            BNE PRENIB1
78            LDX #$55          IL Y A 86 NIBBLES DANS LE
                                BUFFER SEC.
79  LB81E     LDA NIBLBUF2,X    CHARGER LES NIBBLES UN PAR
                                UN...
80            AND #$3F          LES NIBBLES SERONT DE LA FORME
                                00XXXXXX
81            STA NIBLBUF2,X    REMET-LE DANS SON BUFFER
82            DEX               IL EN RESTE ?
83            BPL LB81E         OUI, ALORS AU PROCHAIN...
84            RTS
85
86  *================================
87  *WRITE:LA VRAIE ROUTINE QUI ECRIT
88  *      LES NIBBLES DES BUFFERS
89  *      PRIMAIRE ET SECONDAIRE
90  *================================
91
92  WRITE     SEC
93            STX L27           DANS X IL Y A LE NUMERO DE SLOT
                                * 16
94            STX L0678
95            LDA WRLTCH,X      TESTE UNE PROTECTION A
                                L'ECRITURE
96            LDA RDMODE,X
97            BMI LB8B4         SI LE BIT 8 (N) = 1, PROTEGE
98            LDA NIBLBUF2
99            STA COMPTEUR      FIXE LE COMPTEUR DE NIBBLES
100           LDA #$FF          CHARGE UN OCTET DE
                                SELF-SYNCHRONISATION
101           STA WRMODE,X      ECRIT LE SUR LE DISK
102           ORA RDLTCH,X
103           PHA               PERTE DE TEMPS DE 3+4+2=9
                                CYCLES
104           PLA
105           NOP
106
107           LDY #$04          METTRE MAINTENANT 4 SELF-SYNC.
108 LB84A     PHA               FAIT EN DELAI DE 3+4=7 CYCLES
109           PLA
110           JSR WRTNIBL2      PUIS VA ECRIRE
111           DEY               DEMANDE A PASSER AU PROCHAIN
112           BNE LB84A         S'IL EN RESTE...
113
114           LDA #$D5          CHARGE LE DEBUT DU PROLOGUE
                                DANS L'ACC.
115           JSR WRTNIBL1      ET VA ENSUITE L'ECRIRE SUR LE
                                DISK
116           LDA #$AA          $D5/$AA/$AD REPRESENTE LE
                                PROLOGUE DU
117           JSR WRTNIBL1      CHAMP DATA.
118           LDA #$AD
119           JSR WRTNIBL1      L'ECRITURE DU PROLOGUE EST
                                FINIE
120
121           TYA
122           LDY #$56          PREPARE A ECRIRE 86 NIBBLES DU
                                BUF.SEC.
123           BNE LB869         TJS PRIS
124 LB866     LDA NIBLBUF2,Y
125 LB869     EOR NIBLBUF1,Y
126           TAX
127           LDA DATA3,X
128           LDX L27
129           STA WRLTCH,X      WRITE LOAD (WRMODE DOIT AVOIR
                                ETE MIS )
130           LDA RDLTCH,X      ECRIRE L'OCTET (AVANT -> WRITE
                                MODE)
131           DEY               ON LES PREND DE HAUT EN BAS
132           BNE LB866         S'IL EN RESTE A ECRIRE...
133
134           LDA COMPTEUR
135           NOP
136 LB87E     EOR NIBLBUF,Y     LE CODAGE DES NIBBLES A
                                L'ECRITURE
137           TAX
138           LDA DATA3,X
139           LDX L0678
140           STA WRLTCH,X      WRITE LOAD
141           LDA RDLTCH,X      ECRIT L'OCTET
142           LDA NIBLBUF,Y     PREPARE A CODER LE PROCHAIN
                                NIBBLE
143           INY               PREND DU BUFF.PRIMAIRE, DE BAS
                                EN HAUT
144           BNE LB87E         AU PROCHAIN DE CES 256
                                NIBBLES...
145
146           TAX               PREPARE LE CHEKSUM
147           LDA DATA3,X
148           LDX L27
149           JSR WRTNIBL3      ET VA L'ECRIRE...
150
151           LDA #$DE          CHARGE LE DEBUT DE L'EPILOGUE
152           JSR WRTNIBL1      ET VA L'ECRIRE...
153           LDA #$AA          $DE/$AA/$EB EST L'EPILOGUE DU
                                CHAMP.
154           JSR WRTNIBL1
155           LDA #$EB          FINIR L'ECRITURE DE L'EPILOGUE
156           JSR WRTNIBL1
157           LDA #$FF
158           JSR WRTNIBL1
159
160           LDA RDMODE,X      PREPARE POUR DES ENTREES
161 LB8B4     LDA RDLTCH,X      FAIT L'INVERSE
162           RTS
163
164 WRTNIBL1  CLC               LE TEMPS EST IMPORTANT: 32
                                CYCLES
165 WRTNIBL2  PHA               POUR L'ECRITURE D'UN OCTET.
166           PLA
167 WRTNIBL3  STA WRLTCH,X      WRITE LOAD
168           ORA RDLTCH,X      WRITE BYTE
169           RTS
170
171 *================================
172 * POSTNIBL:  DECODE 342 NIBBLES
173 *             (6 BITS) 00XXXXXX EN
174 *             256 OCTETS (8 BITS)
175 *================================
176
177 POSTNIBL  LDY #$00
178 POSTNIB1  LDX #$56          FAITS-EN 85 D'UN COUP...
179 POSTNIB2  DEX
180           BMI POSTNIB1      ENCORE UNE FOIS S.V.P
181           LDA NIBLBUF,Y     DECODE CEUX DU BUFFER PRIMAIRE
182           LSR NIBLBUF2,X    ET CEUX DU SECONDAIRE
183           ROL               SELON LA METHODE DU 6+2
184           LSR NIBLBUF2,X
185           ROL
186           STA (BUFPTR),Y    BUFPTR DONNE L'ADRESSE DE LA
                                PAGE OU
187           INY               SONT RANGES LES OCTETS
                                UTILISABLES
188           CPY COMPTEUR      FAIT 4 FOIS 86 NIBBLES
189           BNE POSTNIB2      OK ?
190           RTS
191
192 *================================
```

```
193  * READ : LIT UN SECTEUR DE DATAS
194  *        DEPUIS LE DISK ET STOCKE
195  *        DANS LES NIBBLE BUFFERS
196  * (LE 2ND D'ABORD DE HAUT EN BAS)
197  * (LE 1ER ENSUITE DE BAS EN HAUT)
198  *================================
199
200  READ       LDY #$20
201  WAITDATA   DEY
202             BEQ RDERR
203  WAITDATA1  LDA RDLTCH,X     LIT UN DATA
204             BPL WAITDATA1
205  WAIT1      EOR #$D5         EN FAIT, ATTEND LE PROLOGUE
                                 D5/AA/AD
206             BNE WAITDATA     CE N'ETAIT PAS SON DEBUT, ALORS
                                 RE-LIT
207             NOP
208  WAIT2      LDA RDLTCH,X     ON ATTEND LE 2EME MORCEAU
209             BPL WAIT2
210             CMP #$AA
211             BNE WAIT1        IL EST FAUX ???
212             LDY #$56         PREPARE LE COMPTEUR A 86 ($56)
213  WAIT3      LDA RDLTCH,X     PUIS LA FIN DE CE PROLOGUE...
214             BPL WAIT3
215             CMP #$AD
216             BNE WAIT1        C'EST ÇA ?
217
218             LDA #$00         LE PROLOGUE EST LU, AUTOUR DES
                                 86 NIB.
219  DATABLOC   DEY              QUI SERONT MIS DANS LE BUFFER
                                 SECOND.
220             STY COMPTEUR     ET RANGES DE HAUT EN BAS
221  DATABL2    LDY RDLTCH,X     LECTURE D'UN DATA DEPUIS LE
                                 DISK
222             BPL DATABL2      JUSQU'A CE QU'IL SOIT
                                 CONVENABLE
223             EOR $BA00,Y
224             LDY COMPTEUR     OU LE RANGE-T-ON CE NIBBLE ?
225             STA NIBLBUF2,Y   A CET ENDOIT DANS LE BUFFER
                                 SECONDAIRE.
226             BNE DATABLOC     IL EN RESTE A LIRE, ALORS
                                 ALLONS-Y
227
228  LB913      STY COMPTEUR
229  LB915      LDY RDLTCH,X     DANS L'ATTENTE D'UN DATA
                                 CONVENABLE
230             BPL LB915        C'EST RATE, IL LUI MANQUE UNE
                                 CASE (!)
231             EOR $BA00,Y
232             LDY COMPTEUR     ON LE RANGERA OU CE NIBBLE ?
233             STA NIBLBUF,Y    ICI DANS LE BUFFER PRIMAIRE
234             INY              OU ILS SERONT STOCKES DE BAS EN
                                 HAUT
235             BNE LB913        ON EN VEUT 256, SOIT 342 AU
                                 TOTAL
236
237  LB925      LDY RDLTCH,X     LECTURE D'UN CHECKSUM
238             BPL LB925        CE BYTE EST TOO MOCHE !
239             CMP $BA00,Y      COMPARE AVEC CELUI CALCULE EN
                                 CHARGEANT
240             BNE RDERR        ERREUR, ILS SONT DIFFERENTS !!!
241
242  LB92F      LDA RDLTCH,X     ON LIT UN DATA...
243             BPL LB92F        BOUH QU'IL EST LAID !
244             CMP #$DE         ...QUI DOIT ETRE LE DEBUT DE
                                 L'EPILOGUE
245             BNE RDERR        ENCORE UNE ERREUR POSSIBLE
246             NOP              HEUREUSEMENT, ÇA SEMBLE ETRE
                                 BON...
247  LB939      LDA RDLTCH,X     ICI, ON S'ATTEND A UN $AA
248             BPL LB939
249             CMP #$AA         C'EN EST UN ? (O/N)
250             BEQ GOODRD       LA LECTURE EST TOUT A FAIT
                                 CORRECTE !!!
251  RDERR      SEC              NON, ALORS ON LE SIGNALE, ET ON
                                 PART...
252             RTS
253
254  *================================
255  * RDADR : LIT LE CHAMP ADRESSE
256  *================================
257
258  RDADR      LDY #$FC         MISE A JOUR DU COMPTEUR
259             STY COMPTEUR
260  LB948      INY
261             BNE WAITDAT
262             INC COMPTEUR
263             BEQ RDERR        IL Y A EU TROP D'ERREUR !!!
264
265  WAITDAT    LDA RDLTCH,X     LECTURE D'UNE DONNEE DU DISK
266             BPL WAITDAT      SEULEMENT SI ELLE N'EST PAS
                                 CONVENABLE
267  WAITD1     CMP #$D5         ÇA DOIT ETRE $D5 POUR LE
                                 PROLOGUE
268             BNE LB948        ERREUR, ON RE-ESSAYE
269             NOP              C'EST BON, ON CONTINUE...
270  WAITD2     LDA RDLTCH,X     LECTURE DU DEUXIEME OCTET D'UN
                                 PROLOGUE
271             BPL WAITD2       LE 8EME BIT DOIT ETRE A 0
272             CMP #$AA         LE PROLOGUE DU CHAMP ADRESSE:
                                 D5/AA/96
273             BNE WAITD1       RE-ESSAYE, MAIS C'EST PAS
                                 ILLIMITE !
274             LDY #$03         C'EST POUR LA PROCHAINE ETAPE
275  WAITD3     LDA RDLTCH,X     CETTE LECTURE DOIT DONNER $96
                                 ET DONC
276             BPL WAITD3
277             CMP #$96         MARQUER LA FIN DU PROLOGUE
278             BNE WAITD1       TOUT EST A REFAIRE...
279
280             LDA #$00
281  ADRBLOCK   STA L27
282  ADRBLOC2   LDA RDLTCH,X     LECTURE D'UN DATA
283             BPL ADRBLOC2     S'IL N'EST PAS VALABLE
284             ROL              DEBUT DU CODAGE...
285             STA COMPTEUR
286  ADRBLOC3   LDA RDLTCH,X     LE VOLUME,PISTE,SECTEUR,
                                 CHECKSUM SONT
287             BPL ADRBLOC3     CODES SUR DEUX OCTETS DANS LE
                                 CHAMP
288             AND COMPTEUR     ADRESSE. D'OU LES 2 LECTURES...
289             STA TEMP,Y       ...SUITE DU CODAGE...
290             EOR L27
291             DEY              IL Y AVAIT $3 POUR LIRE LES 4
                                 DONNEES
292             BPL ADRBLOCK     SI Y=0 VA QUAND MEME LIRE,
                                 ARRETE A $FF
293             TAY
294             BNE RDERR        SI LE DERNIER (CHECKSUM) <> 0,
                                 ERREUR!
295
296  * ATTEND $DE $AA ,L'EPILOGUE, SINON, C=1
297
298  LB98B      LDA RDLTCH,X     MAINTENANT, ON LIT
                                 L'EPILOGUE...
299             BPL LB98B        CET OCTET N'EST PAS DIGESTE
                                 (0XXXXXXX)
300             CMP #$DE         S'IL L'EST, ÇA DOIT ETRE $DE
301             BNE RDERR        SINON, IL Y A ERREUR
302             NOP
303  LB995      LDA RDLTCH,X     EN FAIT, ON NE VA LIRE QU'UN
                                 PARTIE DE
304             BPL LB995        L'EPILOGUE: $DE/$AA/($EB)
305             CMP #$AA
306             BNE RDERR        LE CHAMP ADRESSE N'EST PAS BON
307  GOODRD     CLC              BONNE LECTURE !, PAS D'ERREUR
                                 (C=0)
308             RTS
309
310  *================================
311  * SEEKABS:POSITIONNE LA TETE DE
312  *         LECTURE SUR LA PISTE
313  *         DESIREE
314  *================================
315
316  SEEKABS    STX SLOTNUM      SAUVE LE NUMERO DE SLOT
317             STA TRKN         LA PISTE DE DESTINATION
318
319             CMP CURTRK       ON Y EST DEJA ?
320             BEQ LB9FC        OUI, ALORS C'EST FINI.
321
322             LDA #$00         ON MET A JOUR LE COMPTEUR
323             STA COMPTEUR
324
325  LB9AD      LDA CURTRK       OU ON EST...
326             STA L27
327             SEC              A QUOI ON RETIRE...
328             SBC TRKN         OU ON VA
```

```
329             BEQ LB9EA        IDENTIQUE, ALORS PHASE X OFF
330             BCS LB9C0        ON EST TROP LOIN !
331             EOR #$FF         ON N'EST PAS ASSEZ LOIN
332             INC CURTRK       ALORS PASSE A LA PROCHAINE
                                 PISTE...
333             BCC LB9C5        TJS PRIS
334   LB9C0     ADC #$FE
335             DEC CURTRK       PASSE A LA PRECEDENTE PISTE...
336   LB9C5     CMP COMPTEUR
337             BCC LB9CB
338             LDA COMPTEUR
339   LB9CB     CMP #$0C
340             BCS LB9D0
341             TAY
342   LB9D0     SEC              PHASE X ON, LE BRAS VA BOUGER !
343             JSR LB9EE        VA LE FAIRE CHANGER DE PLACE...
344             LDA DATA1,Y      CHARGE LE DELAI
345             JSR $BA00        ET VA ATTENDRE...
346             LDA L27
347             CLC              C=0 -> PHASE X OFF
348             JSR LB9F1        VA LA METTRE OFF CETTE PHASE
                                 X...
349             LDA DATA2,Y      CHARGE UN DELAI POUR LES
                                 MOUVEMENTS
350             JSR WAITARMOV    DU BRAS, ET EXECUTE-LE...
351             INC COMPTEUR     IL EN RESTE ?
352             BNE LB9AD        IL FAUT ENCORE VERIFIER (TJS
353                              PRIS)
354   * BOUGE LA TETE DE LECTURE
355
356   LB9EA     JSR WAITARMOV    ATTENTE...
357             CLC              C=0 DONC PHASE OFF
358   LB9EE     LDA CURTRK
359   LB9F1     AND #$03         N' PISTE: 000000XX -> NUMERO DE
                                 PHASE
360             ROL              00000XXC (C=0 :OFF , C=1 :ON)
361             ORA SLOTNUM      AJOUTE LE NUMERO DU SLOT
362             TAX
363             LDA PHSOFF,X     MET ON OU OFF LA PHASE REQUISE
364             LDX SLOTNUM      REPREND LE NUMERO DU SLOT
365   LB9FC     RTS
366
367
368   *================================
369   * BOUCLE D'ATTENTE POUR LE
370   * MOUVEMENT DU BRAS EN UNITES DE
371   * 100 MICRO SECONDES
372   *================================
373
374             ORG $BA00
375             OBJ $8A00
376
377
378   WAITARMOV LDX #$11         2 CYCLES
379   WTRMV     DEX              2 CYCLES
380             BNE WTRMV        3 CYCLES -> (2+3)*15+2+2=79
                                 CYCLES
381             INC MONTIME      IL DOIT Y AVOIR LE COMPTEUR DE
                                 TEMPS
382             BNE WTRMV1       QUI SE TROUVE DANS L'IOB (6+3 =
                                 9 CYC.)
383             INC MONTIME+1
384   WTRMV1    SEC              2 CYCLES (92)
385             SBC #$01         4 CYCLES (96)
386             BNE WAITARMOV    SOIT TOUJOURS A LA FIN ENVIRON
                                 100 µSEC
387             RTS
388
389
390   *================================
391   * WRTFORM: ECRIT LE CHAMP ADRESSE
392   * PENDANT L'INITIALISATION.
393   *================================
394
395             ORG $BC56
396             OBJ $8C56
397
398
399   WRTFORM   SEC
400             LDA WRLTCH,X     TESTE S'IL Y A PROTECTION A
                                 L'ECRITURE
401                              LDA RDMODE,X
402   BMI WRIT4                  IL Y EN A UNE, ALORS ARRETE
                                 L'OPERATION
403             LDA #$FF         DEBUT DE L'AUTO-
                                 SYNCHRONISATION...
404             STA WRMODE,X     CELA PERMET D'ECRIRE
                                 L'ACCUMULATEUR
405             CMP RDLTCH,X
406             PHA              L'APPLE VA TROP VITE POUR LE
                                 DRIVE, IL
407             PLA              NE FAUT PAS L'OUBLIER !
408
409   * ECRIT LE PROCHAIN NIBBLE D'AUTO
410   * SYNCHRONISATION. IL Y EN A Y...
411
412   WRIT3     JSR WRITEND      2*12 µS. D'ATTENTE...
413             JSR WRITEND
414             STA WRLTCH,X     WRITE LOAD
415             CMP RDLTCH,X     WRITE BYTE
416             NOP
417             DEY              AU PROCHAIN NIBBLE
418             BNE WRIT3        S'IL EN RESTE...
419
420             LDA #$D5         ECRITURE DU PROLOGUE DU CHAMP
                                 ADRESSE
421             JSR WNIBLB       CONSTITUE DE $D5/$AA/$96
422             LDA #$AA
423             JSR WNIBLB
424             LDA #$96
425             JSR WNIBLB
426             LDA VOL          PUIS COMME L'USAGE LE VEUT, IL
                                 FAUT
427             JSR WBYTE        ECRIRE LE VOLUME, LA PISTE, LE
                                 SECTEUR
428             LDA TRKNO
429             JSR WBYTE
430             LDA SECTNO
431             JSR WBYTE
432
433             LDA VOL          ON FAIT LE CHECKSUM DU VOLUME,
434             EOR TRKNO        DU NUMERO DE PISTE
435             EOR SECTNO       DU NUMERO DE SECT.
436             PHA              ON LE SAUVE POUR USAGE
                                 ULTERIEUR
437             LSR              PUIS ON CONTINUE LE CODAGE
438             ORA BUFPTR       ON A ALORS LE 1ER OCTET DU
                                 CHECKSUM
439             STA WRLTCH,X     QUE L'ON ECRIT...
440             LDA RDLTCH,X
441
442             PLA              REPRENDRE LE RESULTAT
                                 PRECEDEMENT SAUVE
443             ORA #$AA         POUR OBTENIR LE 2EME OCTET DU
                                 CHECKSUM
444             JSR WNIBLA       (TJS 2 OCTETS DANS LE CHAMP
                                 ADRESSE)
445
446             LDA #$DE         PUIS ON FINI ENFIN PAR
                                 L'EPILOGUE
447             JSR WNIBLB
448             LDA #$AA         UN EPILOGUE, C'EST $DE/$AA/$EB
449             JSR WNIBLB
450             LDA #$EB
451             JSR WNIBLB
452             CLC
453   WRIT4     LDA RDMODE,X     ON S'ARRETE D'ECRIRE
454             LDA RDLTCH,X
455   WRITEND   RTS              ET ON RENTRE...
456
457   WBYTE     PHA              ROUTINE QUI ECRIT UN OCTET EN
                                 DOUBLE
458             LSR              CODE LES INFO. ADR. EN BITS
                                 PAIRES ET
459             ORA BUFPTR       IMPAIRES, ET LES ECRIT TOUT LES
                                 32 µSEC
460             STA WRLTCH,X     WRITE LOAD
461             CMP RDLTCH,X     WRITE BYTE
462             PLA
463             NOP              UN NOP PREND 2 CYCLES
464             NOP              DONC ON ATTEND 6 CYCLES...
465             NOP
466             ORA #$AA         $AA = %10101010
467   WNIBLA    NOP
468   WNIBLB    NOP
469             PHA
470             PLA
471             STA WRLTCH,X     ECRIRE LE DEUXIEME OCTET
472             CMP RDLTCH,X
473             RTS
```

Suite de RWTS page 26 ⇒

OFFRES SPECIALES BOUTIQUE IEF
Pour les LECTEURS de POM'S

EXTENSION MAC

Extension **128 K** à **512 K** pour **MAC**

Garantie 1 an I.E.F.

Prix spécial lecteur POM'S 3 362 Frs H.T
3 990 Frs T.T.C

(Nous consulter pour extensions supérieures)

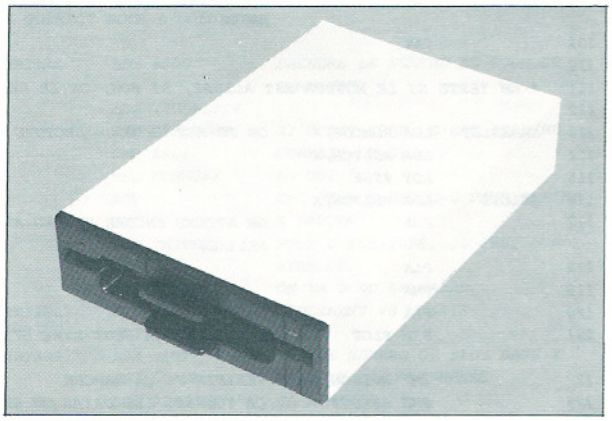
Le meilleur du marché

FLOPPY DISK

5" 1/4 compatible Apple IIe et IIc

CHINON

- 48 T.P.I
- 40 pistes
- 143 Ko sous DOS 3.3
- 160 Ko sous DOS spécial
- ultra silencieux
- détection de piste 0 par photo détecteur
- vitesse ultra stable par moteur à entraînement direct



DRIVE seul : **990 Frs H.T**
1 174 Frs T.T.C

DRIVE + câble : **1 080 Frs H.T**
1 280 Frs T.T.C

DRIVE + câble + boîtier : **1 172 Frs H.T**
1 390 Frs T.T.C

(prix par quantité sur demande)

1/2 hauteur • haute fiabilité • garantie 1 an

Ce lecteur de disquette a été homologué par la FRAMIF, l'Aérospatiale, le club Thomson, etc...

PROMOS : (Urgent quantité limitée)

- Carte 80 colonnes étendue **990** Frs T.T.C
- Disquettes 5"1/4 **9,90** H.T par boîte de 10 (**11,75** Frs T.T.C)
- Moniteurs 12" haute résolution vert **990** Frs T.T.C
- Promo IIc Noel : **Apple IIc** + Moniteur + Souris + Logiciel : **9 990** T.T.C
- Promo MAC Noel : **MAC** étendu à 512 K : **19 990** Frs T.T.C

BOUTIQUE I.E.F. : 403, rue de Vaugirard (porte de Versailles) 75015 PARIS Tél. : 48.28.06.01

Nom: ..... Prénom: ..... Tél: .....
Adresse: .....
Société: ..... Profession: .....

Je passe commande de :

☐ Extension MAC à 3 990 Frs T.T.C ☐ PROMO .....
☐ Floppy Apple modèle .....

Je joins un chèque d'acompte de 20% soit : .....Frs

P11

RWTS 2

(Assembleur BigMac)

```
 1
 2  ********************************
 3  *                              *
 4  *          DISK II             *
 5  *   READ/WRITE TRACK-SECTOR    *
 6  *        ($BD00.BEAE)          *
 7  *                              *
 8  *      © PAR APPLE COMPUTER     *
 9  *                              *
10  *   DESASSEMBLE PAR P. NEVEU   *
11  *                              *
12  ********************************
13
14
15
16            ORG $BD00
17            OBJ $8D00
18
19  *================================
20  *
21  *  LABELS
22  *
23  *================================
24
25  MOTOROFF  EQU $C088       ETEIND LE MOTEUR
26  MOTORON   EQU $C089       ALLUME LE MOTEUR
27  DRV0EN    EQU $C08A       BRANCHE LE DRIVE 1
28  DRV1EN    EQU $C08B       DRIVE 2
29  RDLTCH    EQU $C08C       SWITCH DE LECTURE DATA
30  WRLTCH    EQU $C08D       SWITCH D'ECRITURE DATA
31  RDMODE    EQU $C08E       SWITCH POUR MODE LECTURE
32  WRMODE    EQU $C08F       ECRITURE
33  PHSOFF    EQU $C080       ZONE ALLUMANT OU ETEIGNANT LES 4
                              PHASES
34  *                         ( $C080.C087 )
35
36  PRENIBL   EQU $B800       NIBBLELIZE POUR ECRIRE APRES.
37  WRITE     EQU $B82A       ECRIT LES BUFFERS PRIMAIRE +
                              SECONDAIRE
38  POSTNIBL  EQU $B8C2       DECODE 342 NIBBLES EN 256 OCTETS
39  READ      EQU $B8DC       LIRE UN SECTEUR DU DISK ->NIBBLE
                              BUFFER
40  RDADR     EQU $B944       LIRE LE CHAMP ADRESSE
41  SEEKABS   EQU $B9A0       TETE DE LECTURE SUR LA PISTE
                              DESIREE
42  WAITARMOV EQU $BA00       ATTENTE POUR MOUVEMENTS DU BRAS
43  DSKFORM   EQU $BEAF       ROUTINE DE FORMATAGE
44  DRV0TRK   EQU $0478       TOUT CELA SE TROUVE CACHE DANS L
45  DRV1TRK   EQU $04F8       TROUS DE LA PAGE TEXT
46  SLOT      EQU $05F8       LE NUMERO DE SLOT
47  CURTRK    EQU $0478       LE NUMERO DE LE PISTE ACTUELLE
48  SEEKCNT   EQU $06F8
49  RETRYCNT  EQU $0578       LE COMPTEUR D'ESSAIS
50
51  IOBPL     EQU $48         L'ADRESSE DE LA TABLE IOB
52  MONTIME   EQU $46
53  IOBPH     EQU IOBPL+1
54  PTRSDEST  EQU $3C
55  DEVCTBL   EQU PTRSDEST
56  CSUM      EQU $2C
57  SECT      EQU CSUM+1      LE NUMERO DE SECTEUR
58  TRKN      EQU $2A         PISTE DE DESTINATION
59  DRIVNO    EQU $35         LE NUMERO DE DRIVE
60  TRACK     EQU SECT+1      LE NUMERO DE LA PISTE
61  VOLUME    EQU TRACK+1     LE NUMERO DE VOLUME DU DISK
62  COMPTEUR  EQU $26         POUR LE BUFFER SECONDAIRE
63  IBDERR    EQU $40         ERR! UNE CHOSE ANORMALE S'EST
                              PRODUITE
64  IBVMME    EQU $20         ERR! LES VOLUMES LUS / VOULUS
                              DIFFERENT
65  IBWPER    EQU $10         ERR! DISK PROTEGE CONTRE L'ECRITU
66
67
68  *================================
69  *================================
70
71            STY IOBPL       AVANT D'ENTRER, A&Y POINTENT SUR
                              L'IOB
72            STA IOBPH
73            LDY #$02
74            STY SEEKCNT
75            LDY #$04
76            STY $04F8
77            LDY #1          CHARGE LE NUMERO  DE SLOT
78            LDA (IOBPL),Y
79            TAX
80            LDY #$0F        CHARGE LE NUMERO DU PRECEDENT
                              SLOT, COMPARE
81            CMP (IOBPL),Y
82            BEQ SAMESLOT    C'EST LE MEME, PAS DE PROBLEME
83
84  * ON UTILISE UN NOUVEAU SLOT.
85  * ATTENDRE L'ARRET DU DRIVE.
86  * POUR LE SAVOIR, ON LIT LES
87  * DATAS DU DISK. ILS DOIVENT ETRE
88  * LES MEME PENDANT AU MOINS 9
89  * MICRO-SECONDES
90
91            TXA             SAUVE LE NOUVEAU NUMERO  DE SLOT
92            PHA
93            LDA (IOBPL),Y   CHARGE LE 'VIEUX'
94            TAX
95            PLA
96            PHA
97            STA (IOBPL),Y   SAUVE LE NOUVEAU
98            LDA RDMODE,X    METS-TOI EN MODE DE LECTURE
99  STILLON   LDY #$08        LES DATAS DOIVENT RESTER STABLES
100           LDA RDLTCH,X    PENDANT 96 µSec
101 NOTSURE   CMP RDLTCH,X    LES DONNEES CHANGENT ?
102           BNE STILLON     OUI...
103           DEY
104           BNE NOTSURE     STABLE DEPUIS SUFFISAMENT DE
                              TEMPS ?
105
106 * LE PRECEDENT DRIVE EST ETEINT MAINTENANT
107
108           PLA             REPRENDRE LE NOUVEAU NUMERO DE
                              DRIVE
109           TAX
110
111 * ON TESTE SI LE MOTEUR EST ALLUME, SI NON, ON LE DEMARRE
112
113 SAMESLOT  LDA RDMODE,X    ON SE MET EN MODE LECTURE
114           LDA RDLTCH,X
115           LDY #$08
116 SSLOT1    LDA RDLTCH,X
117           PHA             ON ATTEND ENCORE UN PEU AVANT DE
                              RELIRE
118           PLA
119           PHA
120           PLA
121           STX SLOT        DELAI, ET ÇA PEUT ETRE UTILE PLUS
                              TARD
122           CMP RDLTCH,X    VERIFIE SI ÇA MARCHE
123           BNE SSLOT2      ÇA TOURNE, LES DATAS ON CHANGE!
124           DEY             CONTINUE A ATTENDRE QUE ÇA CHANGE
125           BNE SSLOT1      SI CE N'EST PAS FINI
126 SSLOT2    PHP             SAUVE LE RESULTAT DU TEST
127           LDA MOTORON,X   ALLUME LE MOTEUR
128           LDY #6          COPIE TOUS LES POINTEURS
129 PTRMOV    LDA (IOBPL),Y
130           STA PTRSDEST-6,Y
131           INY
132           CPY #$0A        ILS SONT TOUS COPIES ?
133           BNE PTRMOV
134           LDY #$03
135           LDA (PTRSDEST),Y
136           STA $47
137           LDY #2          MAINTENANT, AU TOUR DES PARAMETRES
138           LDA (IOBPL),Y   DRIVE 1 OU 2 ?
139           LDY #$10        C'EST LE MEME DRIVE QUE CELUI
                              UTILISE
140           CMP (IOBPL),Y   AUPARAVANT ?
141           BEQ OK          OUI, PAS LA PEINE D'ATTENDRE LE
                              MOTEUR
142           STA (IOBPL),Y   MAINTENANT, ON UTILISE CE DRIVE
143           PLP             SIGNALE QU'IL ETAIT ETEINT (TEST)
144           LDY #$00
145           PHP
146 OK        ROR
147           BCC OK1         C=0 : DRIVE 1 UTILISE AVANT
148           LDA DRV0EN,X    ENGAGE LE DRIVE 1
149           BCS DRVSEL      C=1 : DRIVE 2 UTILISE AVANT
```

```
150  OK1        LDA DRV1EN,X     ENGAGE LE DRIVE 2
151  DRVSEL     ROR DRIVNO       SAUVE LE NUMERO DE DRIVE UTILISE
                                 DANS C
152             PLP
153             PHP
154             BNE DRVSEL2
155
156
157             LDY #7           BOUCLE D'ATTENTE
158  DRVSEL1    JSR WAITARMOV
159             DEY
160             BNE DRVSEL1
161             LDX $05F8
162
163  DRVSEL2    LDY #4           CHARGE LA PISTE
164             LDA (IOBPL),Y
165             JSR MYSEEK       ET VAS-Y
166
167             PLP              LE MOTEUR ETAIT ALLUME ?
168             BNE TRYTRK       SI OUI, NE PAS ATTENDRE
169             LDY $47
170             BPL TRYTRK
171
172  * LE MOTEUR ETAIT 'OFF', ATTENDRE QU'IL PRENNE DE LA VITESSE
173
174  MOTOF      LDY #$12         DELAI DE 100 MICROSECONDE A CHAQUE
                                 FOIS
175  CONWAIT    DEY
176             BNE CONWAIT
177             INC MONTIME
178             BNE MOTOF
179             INC MONTIME+1
180             BNE MOTOF
181
182  * LE DISK EST A LA BONNE VITESSE,
183  * ON PEUT LIRE...
184  * SI CE N'EST PAS LA COMMANDE DE
185  * FORMATAGE, LOCALISER LE SECTEUR
186  * CORRECT POUR L'OPERATION
187
188  TRYTRK     LDY #$0C         PRENDRE LE NUMERO DE CODE DE LA
                                 COMMANDE
189             LDA (IOBPL),Y
190             BEQ GALLDONE     SI C'EST ZERO, C'EST TERMINE...
191             CMP #$04         FORMATE LE DISK ?
192             BEQ FORMDSK      EH OUI !
193             ROR              C=1 POUR LIRE, C=0 POUR ECRIRE
194             PHP              A SAUVER
195             BCS TRYTRK2      POUR L'ECRITURE, IL FAUT DES
                                 NIBBLES.
196             JSR PRENIBL      ON VA DONC EN FAIRE...
197  TRYTRK2    LDY #$30         SEULEMENT 48 ESSAIS
198             STY RETRYCNT
199  TRYADR     LDX SLOT         MET LE NUMERO DE SLOT DANS X
200             JSR RDADR        LIRE LE CHAMP ADRESSE
201             BCC RDRIGHT      ÇA A MARCHE !
202  TRYADR2    DEC RETRYCNT     ENCORE UNE ERREUR !
203             BPL TRYADR       HMM ! ÇA PASSE POUR CETTE FOIS...
204  TRYADR3    LDA CURTRK
205             PHA              SAUVE LA PISTE QU'ON VEUT
206             LDA #$60         ON RE-CALIBRE ENCORE !
207             JSR SETTRK
208             DEC SEEKCNT
209             BEQ DRVERR
210             LDA #4
211             STA $04F8
212             LDA #$00         VA SUR LA PISTE 0
213             JSR MYSEEK
214             PLA
215  GOCAL      JSR MYSEEK       ET CETTE FOIS-CI VA A LA PISTE
                                 CORRECTE
216             JMP TRYTRK2
217
218  * LE CHAMP D'ADRESSE EST BIEN LU
219  * S'ASSURER QUE C'EST LA PISTE,
220  * SECTEUR ET LE VOLUME VOULU.
221
222  RDRIGHT    LDY TRACK        ON EST A LA BONNE PISTE ?
223             CPY CURTRK
224             BEQ RTTRK        SI OUI, C'EST BIEN
225
226  * LE DRIVE ETAIT SUR UNE PISTE
227  * DIFFERENTE. ESSAYE DE
228  * RECALIBRER A PARTIR DE CETTE
229  * PISTE.
230
231             LDA CURTRK       SAUVE LA PISTE DE DESTINATION
232             PHA
233             TYA
234             JSR SETTRK
235             PLA
236             DEC $04F8
237             BNE GOCAL        ON VA RECALIBRER
238             BEQ TRYADR3      SINON, ON VA TOUJOURS ICI
239  DRVERR     PLA              ON RETIRE LA PISTE ACTUELLE
240             LDA #IBDERR      ERREUR !
241  JMPTO1     PLP
242             JMP HNDLERR
243  GALLDONE   BEQ ALLDONE
244  FORMDSK    JMP DSKFORM
245
246  * LE DRIVE EST SUR LA BONNE PISTE, VERIFIE LE VOLUME
247
248  RTTRK      LDY #3           C'EST UN DISK ACCEPTABLE ?
249             LDA (IOBPL),Y    CHARGE LE NUMERO DE VOLUME VOULU
250             PHA              SAUVE-LE
251             LDA VOLUME       CHARGE LE VOLUME ACTUEL
252             LDY #$0E         ET SAUVE LE...
253             STA (IOBPL),Y
254             PLA              REPRENDRE LE VOLUME ATTENDU
255             BEQ CORRECTVOL   TOUS LES VOLUMES SONT ACCEPTES
256             CMP VOLUME
257             BEQ CORRECTVOL   C'ETAIT LE BON VOLUME
258             LDA #IBVMME      CHARGE LE CODE DE L'ERREUR
259             BNE JMPTO1       TOUJOURS PRIS
260  CORRECTVOL LDY #5           VERIFIE SI C'EST LE BON SECTEUR
261             LDA (IOBPL),Y
262             TAY
263             LDA $BFB8,Y
264             CMP $2D
265             BNE TRYADR2      C'EST PAS ÇA !
266
267  * CORRECTSECT:
268             PLP
269             BCC WRIT         C=1 : LIRE, C=0 ECRIRE
270             JSR READ         ALORS VA DONC LIRE...
271             PHP              SAUVE LE STATUS DE LA LECTURE
272             BCS TRYADR2      MAUVAISE LECTURE
273             PLP
274             LDX #0           ON VA DECODER TOUS LES NIBBLE POUR
275             STX COMPTEUR     AVOIR DE BEAUX OCTETS UTILISABLES
276             JSR POSTNIBL
277             LDX SLOT         C'EST QUEL SLOT AU FAIT ?
278  ALLDONE    CLC
279             HEX 24           AVEC LE PROCHAIN OCTET ÇA FAIT UN
                                 'BIT'
280  HNDLERR    SEC              INDIQUE UNE ERREUR
281             LDY #$0D         DONNE LUI LE NUMERO D'ERREUR
282             STA (IOBPL),Y
283             LDA MOTOROFF,X   ARRETE LE MOTEUR...
284             RTS
285  WRIT       JSR WRITE        VA ECRIRE LES NIBBLES MAINTENANT
286             BCC ALLDONE      PAS D'ERREUR.
287             LDA #IBWPER      ERREUR ! HORREUR !
288             BCS HNDLERR
289
290  * C'EST LA ROUTINE QUI CHERCHE LA PISTE N DANS LE SLOT X/16
291  * IF DRIVNO<0 THEN DRIVE 0
292  * IF DRIVNO>0 THEN DRIVE 1
293
294  MYSEEK     PHA
295             LDY #$01         C'EST UN DISK A 2 PHASES ?
296             LDA (DEVCTBL),Y
297             ROR              CARRY = NOMBRE DE PHASES
298             PLA
299             BCC SEEK         1 PHASE PAR PISTE !
300             ASL
301             JSR SEEK
302             LSR CURTRK
303             RTS
304  SEEK       STA TRKN         SAUVE LA PISTE DE DESTINATION (*2)
305             JSR XTOY         MET Y = NUMERO DE SLOT
306             LDA DRV0TRK,Y
307             BIT DRIVNO
308             BMI WASDO        SI MOINS, DRIVE 1
309             LDA DRV1TRK,Y
310  WASDO      STA CURTRK       C'EST OU ON SE TROUVE
311             LDA TRKN         ET ÇA, OU ON VA
```

```
312             BIT DRIVNO
313             BMI ISDRV0
314             STA DRV1TRK,Y
315             BPL GOSEEK        TOUJOURS PRIS
316   ISDRV0    STA DRV0TRK,Y
317   GOSEEK    JMP SEEKABS       REMUES-TOI
318   XTOY      TXA               TRANSFERE X/16 DANS Y
319             LSR
320             LSR
321             LSR
322             LSR
323             TAY
324             RTS
325
```

RWTS 3
(Assembleur BigMac)

```
1
2   ********************************
3   *                              *
4   *                              *
5   *            DISK II           *
6   *    FORMATEUR SEIZE SECTEURS  *
7   *         ($BEAF.BFA6)         *
8   *                              *
9   *                              *
10  *     © PAR APPLE COMPUTER.    *
11  *                              *
12  *   DESASSEMBLE PAR P. NEVEU   *
13  *                              *
14  ********************************
15
16
17
18              ORG $BEAF
19              OBJ $8000
20
21  *==============================
22  *
23  *  LABELS
24  *
25  *==============================
26
27  L2D         EQU $2D
28  BUFPTR      EQU $3E           POINTE SUR LA PAGE DE BUFFER DE
                                  TRAVAIL
29  SECTNO      EQU $3F           NUMERO DE SECTEUR
30  VOL         EQU $41           NUMERO DE VOLUME
31  TRKNO       EQU $44           NUMERO DE PISTE
32  SYNCOUNT    EQU $45
33  IOBPL       EQU $48           ADRESSE DE LA TABLE IOB
34
35  RETRYCNT    EQU $0578         COMPTEUR D'ESSAIS
36  WRITE       EQU $B82A         ECRIRE LES NIBBLES DES BUFFERS
37  READ        EQU $B8DC         LIRE UN SECTEUR ET STOCKE LES
                                  NIBBLES
38  RDADR       EQU $B944         LIT LE CHAMP ADRESSE
39
40  * LE NIBBLE BUFFER: $BB00.BC55
41
42  NIBLBUF     EQU $BB00         ADRESSE DE DEBUT DU BUFFER PRIM
43  NIBLBUF1    EQU $BBFF
44  NIBLBUF2    EQU $BC00         ADRESSE DE DEBUT DU BUFFER
                                  SECONDAIRE
45  WRTFORM     EQU $BC56         CHAMP ADRESSE PENDANT
                                  L'INITIALISATION
46  MYSEEK      EQU $BE5A         CHERCHE LA PISTE
47  SETTRK      EQU $BE95
48
49  MOTOROFF    EQU $C088         ETEINT LE MOTEUR
50
51
52  *==============================
53  *==============================
54
55  DSKFORM     LDY #$03          PREND LE VOLUME DESIRE
56              LDA (IOBPL),Y
57              STA VOL           STOCKE-LE ICI
```

```
326   SETTRK    PHA
327             LDY #$02          CHARGE LE NUMERO DE DRIVE
328             LDA (IOBPL),Y
329             ROR               MIS DANS CARRY
330             ROR DRIVNO        DANS DRIVNO
331             JSR XTOY
332             PLA
333             ASL
334             BIT DRIVNO
335             BMI ONDRV0
336             STA DRV1TRK,Y
337             BPL SETRTS
338   ONDRV0    STA DRV0TRK,Y
339   SETRTS    RTS
```

```
58              LDA #$AA
59              STA BUFPTR
60
61              LDY #$56          NETTOIE LE BUFFER SECONDAIRE
                                  D'ABORD
62              LDA #$00          ON COMMENCERA PAR LA PISTE 0
63              STA TRKNO
64  CLNIBUF     STA NIBLBUF1,Y    REMPLIR DE $00...
65              DEY               DE HAUT EN BAS POUR LE SECOND
66              BNE CLNIBUF       PASSE AU PROCHAIN...
67  CLNIBUF1    STA NIBLBUF,Y     PUIS FAIT LE PRIMAIRE

68              DEY               EGALEMENT...
69              BNE CLNIBUF1      JUSQU'A CE QUE ÇA SOIT TERMINE.
70
71              LDA #$50          $50 = 80
72              JSR SETTRK
73              LDA #$28          $28 = 40, NOMBRE DE SYNC.BYTES
74              STA SYNCOUNT
75
76  * VA CHERCHER LA PISTE DESIREE
77
78  TRKFORM     LDA TRKNO         PRENDRE LE NUMERO DE LA PISTE
                                  DESIREE
79              JSR MYSEEK        SE METTRE AU DESSUS
80              JSR TRKWRT        VA FORMATER LA PISTE EN QUESTION
81              LDA #$08          CHARGE UN CODE D'ERREUR AU CAS
                                  OU...
82              BCS ERROR         S'IL Y A EU UNE ERREUR (C=1)
83              LDA #$30          48 ESSAIS SONT PERMIS
84              STA RETRYCNT      SE LE RAPPELER !
85  LBEE5       SEC               C=1 AU CAS OU ON NE POURRAIT
                                  CONTINUER
86              DEC RETRYCNT      TENTE DE RE-ESSAYER
87              BEQ ERROR         MAIS C'EN EST TROP
88              JSR RDADR         SINON, LIT LE CHAMP D'ADRESSE
89              BCS LBEE5         RE-ESSAYER SI C'EST RATE
90              LDA L2D
91              BNE LBEE5         ERREUR ?!
92              JSR READ          PUIS LIT LE RESTE DES DATAS
93              BCS LBEE5         SI C'EST RATE
94              INC TRKNO         PASSE A LA PROCHAINE PISTE
95              LDA TRKNO
96              CMP #$23          34 PISTES DEJA ?
97              BCC TRKFORM       NON, ON CONTINUE
98              CLC               SI, ALORS LE DISK EST FINI, SANS
                                  ERREUR
99              BCC DSKDONE       TJS PRIS
100 ERROR       LDY #$0D          MET LE CODE D'ERREUR
101             STA (IOBPL),Y     DANS LA TABLE IOB POUR ETUDE
                                  ULTERIEURE
102             SEC               ET SIGNALE LA EN METTANT C=1
103 DSKDONE     LDA MOTOROFF,X    ETEINT LE MOTEUR
104             RTS               ET RETOURNER D'OU ON VIENT...
105
106
107 TRKWRT      LDA #$00          COMMENCE AU 1ER SECTEUR DE LA PISTE
108             STA SECTNO        L'ENREGISTRER
109             LDY #$80          128 SELF.SYNC A ECRIRE
110             BNE LBF17         TJS PRIS
111 LBF15       LDY SYNCOUNT
112 LBF17       JSR WRTFORM       VA ECRIRE LE CHAMP D'ADRESSE
113             BCS LBF87         ERREUR !!
114             JSR WRITE         VA ECRIRE LE CONTENU DES BUFFERS A
                                  NIB.
115             BCS LBF87         HORREUR !!
116             INC SECTNO        PROCHAIN SECTEUR
```

```
117             LDA SECTNO      ON EN EST OU EXACTEMENT ?
118             CMP #$10        16 SECTEURS DEJA ?
119             BCC LBF15       PAS ENCORE, ON CONTINUE
120
121             LDY #$0F        IL Y A 16 SECTEURS A MARQUER
122             STY SECTNO
123             LDA #$30        48 ESSAIS MAXIMUM SONT AUTORISES
124             STA RETRYCNT    STOCKE-LE POUR NE PAS L'OUBLIER
125     LBF32   STA CARTE,Y     Y METTRE DES $30 AUSSI...
126             DEY             16 EN TOUT
127             BPL LBF32       PAS UN DE MOINS !
128
129             LDY SYNCOUNT    ATTENDRE SELON SYNCOUNT
130     LBF3A   JSR LBF87       12 CYCLES
131             JSR LBF87       +12 CYCLES
132             JSR LBF87       +12 CYCLES
133             PHA             + 3 CYCLES
134             PLA             + 4 CYCLES
135             NOP             + 2 CYCLES
136             DEY             + 2 CYCLES
137             BNE LBF3A       + 4 SI PRIS, SOIT 51 AU TOTAL PAR
                                TOUR
138
139             JSR RDADR       VA LIRE LE CHAMP ADRESSE
140             BCS LBF71       SI IL Y A UNE ERREUR
141             LDA L2D         LECTURE CORRECTE, ON EST AU
                                SECTEUR 0 ?
142             BEQ VERIFY      OUI, ALORS VERIFIE
143             LDA #$10        NON.
144             CMP SYNCOUNT
145             LDA SYNCOUNT
146             SBC #$01
147             STA SYNCOUNT
147             CMP #$05
149             BCS LBF71
150             SEC
151             RTS
152
153     VRFYTRK JSR RDADR       VERIFIE LE CHAMP ADRESSE
154             BCS VRFERR      IL Y A EU UNE ERREUR
155     VERIFY  JSR READ        VERIFIE LE CHAMP DATA
156             BCC SCTMAP      PAS D'ERREUR, SAUTE PAR DESSUS
157     VRFERR  DEC RETRYCNT    RE-ESSAYE DEPUIS LE DEBUT
157             BNE VRFYTRK
159     LBF71   JSR RDADR       LIRE LE CHAMP ADRESSE
160             BCS VRFERR1     ERREUR SUR LE CHAMP ADRESSE !
161             LDA L2D
162             CMP #$0F
163             BNE VRFERR1
164             JSR READ        ET VA LIRE LE CHAMP DATA
165             BCC TRKWRT      IL N'Y A PAS D'ERREUR, SAUTE...
166     VRFERR1 DEC RETRYCNT
167             BNE LBF71       ON PEUT RE-ESSAYER
168             SEC             PLUS D'ESSAI A FAIRE, C'EST RATE
169     LBF87   RTS             ALORS RENTRE A LA MAISON
170
171     * ROUTINE QUI MARQUE LA CARTE DES
172     * SECTEURS INITIALISES LORSQU'UN
173     * SECTEUR EST VERIFIE
174
175     SCTMAP  LDY L2D
176             LDA CARTE,Y     CHARGE POUR VOIR...
177             BMI VRFERR      IL Y A ERREUR, C'EST NEGATIF !
178             LDA #$FF        C'EST BON, ON VA Y METTRE $FF
179             STA CARTE,Y     VOILA, C'EST FAIT
180             DEC SECTNO      AU PROCHAIN SECTEUR
181             BPL VRFYTRK     ET VERIFIE LES TOUS
182             LDA TRKNO       ON EN EST A QUELLE PISTE ?
183             BNE LBFA6       PAS A LA DERNIERE EN TOUT CAS
184             LDA SYNCOUNT
185             CMP #$10
186             BCC LBF87
187             DEC SYNCOUNT    SI SYNC.COUNT > 16
188             DEC SYNCOUNT
189     LBFA6   CLC             C'EST FINI, ET SANS ERREUR
190             RTS
191
192     * CETTE CARTE COMMENCE EN $BFA8
193     * ET CONTIENT LES SECTEURS
194     * INITIALISES:
195     * $30 AVANT, $FF APRES.
196     * ( $BFA8.BFB7 )
197
198     CARTE   EQU *
199
200             LST OFF
201
202     ********************************
203     *                              *
204     * SOURCES:                     *
205     *                              *
206     * -BENEATH APPLE DOS           *
207     * -POM'S 9                     *
208     * -CALL APPLE : ALL ABOUT DOS  *
209     *                              *
210     ********************************
```


Vous avez un Apple //e avec Chat Mauve ou un //c ?
Vous disposez de Pascal 1.2 ?

Profitez de l'interpréteur **COGO**

Par Nicolas Montsarrat

Il s'agit d'un système graphique double-haute résolution écrit en Pascal.
COGO vous permet de manipuler des graphiques grâce à un langage de description des objets - points, angles- et à l'emploi de fonctions primitives de manipulation très puissantes : cercle, tangente, intersections, parallèles, etc. Il est ainsi possible de tracer des grilles, des cercles, des segments de droite, des tangentes communes à deux cercles, de calculer des distances, des angles...

L'éditeur permet une saisie rapide du langage. Une instruction COGO peut-être exécutée dès la saisie pour faciliter la mise au point, ou au sein d'un programme.

Ce programme, destiné à résoudre des problèmes de géométrie plane, comporte des instructions de stockage sur fichier afin de permettre la reprise d'un calcul.

NB : Sur l'Apple //c, l'affichage se fait en simple haute résolution.

150,00 F TTC, franco
Bon de commande page 74

Ce programme vous permet de décrypter des secteurs codés. Il contient différents algorithmes qui ont chacun pour fonction, d'une manière différente, d'effectuer le décodage d'informations cryptées. Il vous offre ainsi la possibilité de voir afficher en clair du texte codé mémorisé à un emplacement donné. Ceci est particulièrement utile, par exemple pour voir afficher les textes de nombreux jeux d'aventure, dont les réponses sont la plupart du temps cryptées par les auteurs du logiciel.

Décrypteur

Paul Fransceschi

Attention : cette fonction ne vous est procurée que pour vous permettre de profiter au maximum de vos logiciels d'aventure, en permettant d'aller plus avant grâce à l'aide fournie par le décodage. Au lieu d'en rester aux trois premières images du jeu, vous pouvez aller beaucoup plus loin et profiter bien plus pleinement de votre logiciel d'aventure. En aucun cas cependant, vous ne pouvez utiliser cette capacité de décryptage en portant atteinte aux droits des auteurs du logiciel que vous êtes parvenu à décrypter.

Il existe bien entendu d'autres utilisations possibles de ce logiciel dont le mode précédent ne constitue qu'un exemple : il vous est ainsi loisible d'appliquer des techniques de décodage à n'importe quel texte, qu'il s'agisse de fichiers, de programmes, ou d'informations quelconques. Vous pouvez également vous livrer au plaisir du décodage, en tant qu'exercice intellectuel.

D'autre part, le programme vous permet également d'effectuer des vérifications sur des informations que vous avez préalablement codées, et de vous assurer que le cryptage est correct.

Une fonction particulière permet par ailleurs de voir afficher très rapidement la totalité des caractères d'un secteur, sur l'écran de l'Apple; il est ainsi possible d'obtenir une vue d'ensemble de ce qui se trouve sur la totalité de la disquette; ceci vous permet notamment de déterminer à quels emplacements se trouvent les informations cryptées et celles qui ne le sont pas.

Mode d'emploi

Au début, vous choisissez une piste et un secteur à décrypter; lorsque cette sélection est faite, vous entrez immédiatement dans l'écran de décryptage, qui offre une vision complète des 256 caractères présents dans le secteur choisi.

Note : si le code du caractère est celui d'un caractère de contrôle, un point est affiché à la place du caractère.

A droite et en haut du texte apparaît le filtre, élément fondamental du décryptage que vous allez ensuite appliquer au texte, et qui se compose de trois parties : octet 1, octet 2 et octet 3.

En bas et à droite se trouvent deux mémoires, MEM1 et MEM2, qui seront ensuite utilisées par certains algorithmes.

En bas et à gauche de l'écran apparaît la commande, en caractères inversés. Vous sélectionnez les commandes à l'aide des touches "flèche à droite", "flèche à gauche", et "Return".

Faites votre choix à l'aide des deux flèches, et validez la commande choisie avec un "Return".

Programme DECR.SOURCE (Assembleur Lisa 2.5)

NDLR: les numéros de ligne non affectés correspondent à des lignes vides ("44 ; ", par exemple).

```
  1          NLS
  2 WNDLFT   EPZ $20
  3 WNDWDTH  EPZ WNDLFT+$1
  4 WNDTOP   EPZ WNDWDTH+$1
  5 WNDBTM   EPZ WNDTOP+$1
  6 CH       EPZ $24
  7 CV       EPZ CH+$1
  8 CKSW     EPZ $38
 10 SPDBYT   EPZ $F1
 11 WARMST   EQU $3D0
 12 COLDST   EQU $3D3
 13 INTDOS   EQU $3EA
 14 SOFTEV   EQU $3F2
 15 IRQLOC   EQU $3FE
 16 KBD      EQU $C000
 17 KBDSTRB  EQU $C010
 18 SPKR     EQU $C030
 19 TXTCLR   EQU $C050
 20 TXTSET   EQU TXTCLR+$1
 21 MIXCLR   EQU TXTSET+$1
 22 LOWSCR   EQU MIXCLR+$1
 23 MISCR    EQU LOWSCR+$1
 24 LORES    EQU MISCR+$1
 25 HIRES    EQU LORES+$1
 26 STROUT   EQU $DB3A
 27 RND      EQU $EFAE
 28 PRNTYX   EQU $F940
 29 PRNTAX   EQU $F941
 30 PRBLK2   EQU $F94A
 31 BASCALC  EQU $FBC1
 32 BS       EQU $FC10
 33 VS       EQU $FC1A
 34 VTAB     EQU $FC22
 35 HOME     EQU $FC58
 36 LF       EQU $FC66
 37 RDKEY    EQU $FD0C
 38 KEYIN    EQU $FD1B
 39 PRBYTE   EQU $FDDA
 40 COUT     EQU $FDED
 41 SETINV   EQU $FE80
 42 SETNORM  EQU $FE84
 43 OUTPORT  EQU $FE95
 46 ;
 47 BELL     EQU $87
 48 LARROW   EQU $88
 49 RETURN   EQU $8D
 50 RARROW   EQU $95
 51 ESC      EQU $9B
 52 CTRLZ    EQU $9A
 55 ;
 56 ORA      EQU $09
 57 EOR      EQU $49
 58 AND      EQU $29
 59 ASL      EQU $0A
 60 LSR      EQU $4E
 62 ;
 63          JSR INITVECT
 64 INTRO    JSR HOME
 65          JSR PRINT
 66          HEX 8D8D8D
 67          ASC "PROGRAMME DE DECRYPTAGE"
 68          HEX 8D
 69          ASC "------------------------"
 70          HEX 8D
 71          ASC "DE SECTEURS  CODES PAR "
```

Les différentes fonctions

Pour décrypter, l'outil dont vous devrez user judicieusement est le *"filtre"*. Il faut spécifier certaines valeurs dans les trois octets qui le composent; dans certains cas, ceci est cependant réalisé directement par le programme; aucune connaissance du langage machine n'est requise pour l'utilisation des fonctions principales. Certaines fonctions, lorsqu'elles sont sélectionnées, agissent directement sur le filtre, et la modification effectuée sur ce dernier est affichée instantanément à l'écran.

Ainsi, le "ou logique", le "et logique" et le "ou exclusif", parmi les commandes, agissent directement sur le filtre : le code opération en langage machine de l'opération choisie s'inscrit dans l'octet 1 du filtre.

Pour agir directement sur le filtre, vous disposez de six fonctions accessibles par les commandes; en général, vous n'agirez directement que sur l'octet 2 du filtre; la modification des octets 1 et 3 du filtre est dangereuse et ne vous sera utile que si vous avez des connaissances suffisantes en langage machine.

Voici les six commandes qui agissent directement sur le filtre :

- Incrémenter octet 1 : ajoute 1 en hexadécimal à l'octet 1 du filtre;
- Décrémenter octet 1 : enlève 1 en hexadécimal à l'octet 1 du filtre;
- Incrémenter octet 2 : ajoute 1 à l'octet 2 du filtre;
- Décrémenter octet 2 : enlève 1 à l'octet 2 du filtre;
- Incrémenter octet 3 : ajoute 1 à l'octet 3 du filtre;
- Décrémenter octet 3 : enlève 1 à l'octet 3 du filtre.

Dans tous ces cas, vous devez sélectionner la commande voulue, puis la valider avec "Return". En maintenant constamment la pression sur la touche "Return", on obtient une incrémentation ou décrémentation très rapide.

OU logique

Cet algorithme réalise un "ou logique" avec la valeur spécifiée dans l'octet 2 du filtre, pour chacun des 256 octets du secteur sélectionné. La valeur 09, qui est le code-opération du "ORA" en langage-machine, est placée dans l'octet 1 du filtre.

ET logique

Ce deuxième algorithme réalise un "et" avec la valeur spécifiée dans l'octet 2 du filtre. La valeur 29, code-opération du "AND", se trouve dans l'octet 1 du filtre.

OU exclusif

Idem. Réalise un "ou exclusif" avec la valeur placée dans l'octet 2 du filtre. Le code-opération du "EOR", 49, est placé dans l'octet 1 du filtre.

Echange de bits

Cet autre algorithme effectue un échange entre deux bits donnés de chaque octet parmi les 256 du secteur choisi. On précise dans MEM1 et MEM2 une valeur entre 0 et 7, qui est le rang des 2 bits dont on veut effectuer la permutation. Avant d'utiliser cet algorithme, vous devez donc spécifier deux valeurs différentes, comprises entre 0 et 7, pour les deux variables utilisées par l'algorithme, MEM1 et MEM2. Pour modifier ces valeurs, utilisez les commandes suivantes :

- Incrémenter MEM1 : ajoute 1 en hexadecimal à la valeur actuelle de MEM1.
- Décrémenter MEM1 : enlève 1 à MEM1.
- Incrémenter MEM2 : ajoute 1 à MEM2.
- Décrémenter MEM2 : enlève 1 à MEM2.

Lire secteur

Cette commande permet de lire et d'afficher le contenu d'un nouveau secteur dont vous spécifiez les coordonnées. Un filtre neutre (EA EA EA) est appliqué à ce secteur, qui apparaît tel qu'il est mémorisé.

Exécution

Cette commande réalise l'affichage des 256 caractères du secteur choisi, en lui appliquant le filtre courant.

```
 72          HEX 8D
 73          ASC "UTILISATION D'UN FILTRE"
 74          HEX 8D
 75          ASC "------------------------"
 76          HEX 8D8D8D
 77          ASC "     PAR PAUL FRANCESCHI"
 78          HEX 8D
 79          ASC "     COPYRIGHT 1985."
 80          INV "VERSION 1.1"
 81          HEX 00
 84 ;
 85 TEST0    LDA KBD
 86          BPL TEST0
 87          BIT KBDSTRB
 88 ;
 89 ECR2     JSR HOME
 90          JSR PRINT
 91          ASC "CE PROGRAMME VOUS
                  PERMET D'AFFICHER LE"
 92          HEX 8D
 93          ASC "CONTENU D'UN SECTEUR
                 DE MULTIPLES  FA-"
94          HEX 8D
95          ASC "CONS, EN COMMUTANT A
                 CHAQUE  FOIS  UN"
96          HEX 8D
97          ASC "FILTRE, MATERIALISE
                 PAR TROIS OCTETS."
98          HEX 8D
99          HEX 8D
100         ASC "VOUS POUVEZ AINSI
                 VISUALISER UN  SECT-"
101         HEX 8D
102         ASC "EUR AVEC UN FILTRE
                 CLAIR, FONCE,  OPA-"
103         HEX 8D
104         ASC "QUE OU ORANGE; VOTRE
                 VISION PEUT  ETRE"
105         HEX 8D
106         ASC "FLOUE, PRECISE, DE
                 HAUT, DE PRES , DE "
107         HEX 8D
108         ASC "BIAIS, OU EXTRALUCIDE..."
109         HEX 8D8D8D
110         ASC "ALORS, QUEL SECTEUR
                 DESIREZ-VOUS LIRE?"
111         HEX 8D8D8D
112         INV "  PISTE     "
113         HEX 8D8D
114         INV "  SECTEUR   "
115         HEX 00
117 ;
118         LDA #!14
119         STA INP2CH
120         LDA #!15
121         STA INP2CV
122         JSR ASCONV2
123         LDA ASCV2RES
124         STA IBTRK
125 ;
126         LDA #!14
127         STA INP2CH
128         INC INP2CV
129         INC INP2CV
130         JSR ASCONV2
131         LDA ASCV2RES
132         STA IBSECT
134 ;
135         LDA #!10
136         STA CH
137         LDA #!22
138         STA CV
139         JSR VTAB
140         JSR PRINT
141         BLK "ESPACE POUR CONTINUER"
142         HEX 00
146 ;
147 TEST    LDA KBD
```

Piste suivante

Cette option affiche les 256 caractères du secteur de la piste suivant la piste actuelle; le numéro actuel du secteur est conservé. Le filtre reste le même. Cette option permet d'aller très rapidement de piste en piste, et de passer ainsi en revue toute la disquette. En maintenant constamment le doigt sur la touche "Return", vous effectuez un parcours à grande vitesse de la totalité de la disquette.

Secteur suivant

Le secteur suivant est lu, en conservant le filtre courant. Ceci permet d'appliquer un même filtre à une série de secteurs, sans avoir de manipulation à effectuer.

La démarche à suivre pour effectuer un décodage est généralement la suivante :

- on réalise le décryptage d'un secteur précis à l'aide d'un filtre donné;
- lorsque ce filtre a fonctionné, on l'applique à tous les secteurs suivants.

Ctrl-Z

A tout moment, vous pouvez effectuer une copie sur imprimante des caractères qui sont affichés à l'écran, avant ou surtout après décodage. Combinée avec l'option "Secteur suivant", cette commande permet de réaliser des listings de secteurs décodés, de façon très rapide et commode.

Espace

En appuyant sur la barre d'espacement, une incrémentation automatique de l'octet 2 du filtre est réalisée, en même temps que le filtre nouveau est appliqué aux 256 octets du secteur choisi. Cette commande rend possible un test très rapide des différentes valeurs susceptibles d'être utilisées par les algorithmes "ou exclusif", "et logique", "ou logique". Elle permet de tester les 256 valeurs (00 à FF) que peut prendre l'octet 2 du filtre, de façon très rapide et sans manipulation.

Ctrl-Reset

Ramène au début du programme.

Pour commencer

Si vous utilisez pour la première fois ce programme, voici la manière la plus simple de procéder.

Sélectionnez une piste et un secteur au hasard et réalisez-en l'affichage.

A l'aide de la touche "flèche à droite", sélectionnez l'option "Piste suivante", et validez-la avec "Return".

Appuyez sans arrêt sur "Return" pour obtenir une vision rapide du contenu de la disquette. Votre but est de trouver les emplacements du texte codé : vous les reconnaîtrez en facilement, car le texte codé ne présente en principe jamais plus de deux caractères consécutifs identiques !

Lorsque vous avez trouvé un secteur répondant à cette définition, vous pouvez supposer que vous êtes en présence de texte crypté.

La démarche consiste alors à effectuer le décodage du secteur supposé crypté. Vous allez utiliser à cet effet tous les algorithmes.

Commencez par sélectionner le "ou logique", à l'aide de la touche flèche à gauche, et validez avec "Return"; ensuite, il vous suffit d'appuyer sur la barre d'espacement pour obtenir un test du filtre avec la variable qu'il utilise, l'octet 2. Vous démarrez avec l'octet 2 à la valeur EA; faites le test pour les 256 valeurs possibles de l'octet 2.

Si aucun texte n'est encore apparu, faites la même opération avec le "et logique", puis le "ou exclusif". Si cela ne marche toujours pas, testez avec l'algorithme d'échange de bits, en utilisant toutes les valeurs possibles.

Si vous parvenez à afficher un texte, c'est que l'algorithme a réussi; dans ce cas, revenez à la valeur précédente de l'octet 2 (car la barre d'espacement incrémente systématiquement son contenu) à l'aide de la commande "décrémenter octet 2"; placez-vous ensuite sur l'option "secteur suivant" et

```
148          BPL TEST
149          BIT KBDSTRB
150 ;
151          JSR HOME
152          JSR PRINT
153          ASC "        "
154          ASC "******************"
155          HEX 8D
156          ASC "        "
157          ASC "* DECRYPTEUR 1.0 *"
158          HEX 8D
159          ASC "        "
160          ASC "******************"
161          HEX 00
162          JSR AFFTRSC
163          JMP AFFS
164 ;
165 AFFTRSC  LDA #4
166          STA CV
167          JSR VTAB
168          LDA #0
169          STA CH
170          JSR PRINT
171          ASC "PISTE "
172          HEX 00
173          LDA IBTRK
174          JSR PRBYTE
175          JSR PRINT
176          ASC "  SECTEUR "
177          HEX 00
178          LDA IBSECT
179          JSR PRBYTE
180          RTS
182 ;
183 AFFS     LDA #0
184          STA COMPT
185          STA BUF
186          LDA #$60
187          STA BUF+$1
188 ;
189          JSR RWTS
190          JSR AFF.256
192 ;
193          LDA #!33
194          STA CH
195          LDA #2
196          STA CV
197          JSR VTAB
198          JSR PRINT
199          ASC "FILTRE"
200          HEX 00
201          LDA #!33
202          STA CH
203          LDA #3
204          STA CV
205          JSR VTAB
206          JSR PRINT
207          ASC "------"
208          HEX 00
212 ;
213 ;INITIALISE LE FILTRE ET L'AFFICHE
214          LDA #$EA
215          STA FIL1
216          STA FIL2
217          STA FIL3
218          JSR AF.FIL
221 ;
222          LDA #!33
223          STA CH
224          LDA #!20
225          STA CV
226          JSR VTAB
227          JSR PRINT
228          ASC "M.1 M.2"
229          HEX 00
230 ;
231          JSR AF.MEM
233 ;
```

vous pourrez visualiser les secteurs voisins avec ce filtre efficace. A chaque affichage, vous pouvez effectuer une copie du texte décodé avec "Ctrl-Z".

Pour continuer

La liberté de manipulation du filtre qui vous est laissée vous permet de spécifier un "JSR" à un sous-programme de décryptage qui vous est propre, et tiendra lieu de filtre; placez pour cela le code-opération du "JSR" dans l'octet 1 du filtre, et son adresse dans les octets 2 et 3, dans l'ordre partie basse-partie haute.

Insérez votre routine personnelle de décodage à partir de l'adresse $2000, en veillant à restituer les registres en fin de sous-programme.

A titre d'exemple, nous avons trouvé les filtres permettant de lire les dictionnaires de **Paranoïak**, **Incredible Hulk**, **Critical Mass**. Vous comprendrez qu'il est difficile dans donner la liste dans ces colonnes.

NB : Dans tous les cas, vous avez accès au vocabulaire du jeu qui n'est généralement pas codé, grâce à l'éditeur de secteur, et il est très simple d'en obtenir une impression avec Ctrl-Z.

```
234          LDA #0
235          STA OPIND
236          STA OPCH
237          LDA #!20
238          STA OPCV
239          LDA #!18
240          STA OPINDMAX
241          JSR OPAFF
242          JMP OPTEST
247 ;
248 AFF.256  LDA #0
249          STA CH
250          STA COMPT
251          LDA #8
252          STA CV
253          JSR VTAB
254          LDY #0
255 AFSUIV   LDA (BUF),Y
256          NOP
257 FIL1     NOP
258 FIL2     NOP
259 FIL3     NOP
260          CMP #$80
261          BLT >2
```

Suite du source page 34 ⇒

Une nouvelle disquette Pom's :

P-Format][

Sur Apple][+ 48 Ko ou //e avec ImageWriter

Par J. Gouzy

Ce programme s'utilise en complément d'un éditeur de texte, AppleWriter par exemple. Les textes à imprimer sont enrichis de codes de contrôle destinés à P-Format][, puis sauvegardés sur disquette en format texte. P-Format lit ce fichier et pilote l'imprimante. De nouvelles possibilités sont alors disponibles :

Justification :

Droite, Gauche, Totale, Centrée, Page paire à droite et impaire à gauche de façon automatique. La justification n'est pas perturbée par les caractères du type ê ou ö.

Polices de caractères :

Changement simplifié pour passer du é français au { américain ou ß allemand par exemple.

Style d'impression :

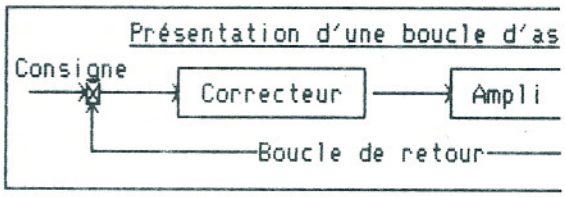
Simple ou double largeur, Gras ou non, Souligné ou non, en mode proportionnel, au sein d'une même ligne sans perte de la justification.

On peut obtenir en justification totale, des caractères **l a r g e s**, des caractères **gras**, ainsi que des caractères souligné**s**.

Mode Mathématique :

$$\sum_{i=1}^{5} \alpha^2 F_i(x,y) = \sqrt{\cos(3.x + y)} = \frac{\int_{-1}^{\beta(x+\delta y)} \frac{\partial \Gamma}{\partial t}.dt}{\lim_{\varepsilon \to 3} \Delta(x+\varepsilon, y-\varepsilon)}$$

Mode graphique:



...tabulation, entête, numérotation des pages...

P-Police][

Sur Apple //e 128 Ko

Sur la même disquette, ce programme vous permet d'éditer en double haute résolution les polices de caractères personnalisées de l'ImageWriter.

P-Format][et P-Police][: 200,00F avec documentation
Bon de commande page 74

```
262          CMP #$A0
263          BGE >2
264 ;AFFICHE UN POINT
265 ;SI CARACT DE CONTROLE
266 ;
267          LDA #$AE
268 ;
269 ^2       JSR COUT
270          INY
271          INC COMPT
272          LDA COMPT
273          CMP #!32
274          BNE AFSUIV
275          LDA #0
276          STA COMPT
277          LDA #0
278          STA CH
279          INC CV
280          JSR VTAB
281          CPY #0
282          BNE AFSUIV
283          JSR SOUND1
284          LDA FIL1
285          STA PFIL1
286          LDA FIL2
287          STA PFIL2
288          LDA FIL3
289          STA PFIL3
290          RTS
293 ;
294 BUF      EPZ $6
295 COMPT    BYT 0
296 NUMERO   BYT 0
297 ;
298 SAVA     BYT 0
299 SAVX     BYT 0
300 SAVY     BYT 0
315 ;
316 RWTS     LDA #DEVTPC
317          STA IBDCTP
318          LDA /DEVTPC
319          STA IBDCTP+$1
320          LDA /IOB
321          LDY #IOB
322          JSR $3D9
323          RTS
324 ;
325 IOB      BYT 1,$60
326 IBRVN    BYT 1,0
327 IBTRK    BYT 0
328 IBSECT   BYT 0
329 IBDCTP   BYT 0,0
330 IBUFFP   BYT 0,$60,0,0
331 IBCMD    BYT 1
332 IBSTATT  BYT 0,0,$60,1
333 DEVTPC   BYT 0,1,$EF,$D8
335 ;
336 AF.FIL   LDA #5
337          STA CV
338          JSR VTAB
339          LDA #!32
340          STA CH
341          LDA FIL1
342          JSR PRBYTE
343          LDA #$A0
344          JSR COUT
345          LDA FIL2
346          JSR PRBYTE
347          LDA #$A0
348          JSR COUT
349          LDA FIL3
350          JSR PRBYTE
351          RTS
353 ;
354 ADRL     EPZ $8
355 ADRH     EPZ ADRL+1
360 ;
361 PRINT    PHP
362          PHA
363          TYA
364          PHA
365          TXA
366          PHA
367 ;
368          TSX
369          LDA $105,X
370          STA ADRL
371          LDA $106,X
372          STA ADRH
373          LDY #1
374 ;
375 PRNTLOOP LDA (ADRL),Y
376          BEQ PRNTDONE
377          JSR $FDED
378          INC ADRL
379          BNE PRNTLOOP
380          INC ADRH
381          BNE PRNTLOOP
382 ;
383 PRNTDONE INC ADRL
384          BNE PRNDN0
385          INC ADRH
386 ;
387 PRNDN0   LDA ADRL
388          STA $105,X
389          LDA ADRH
390          STA $106,X
391 ;
392          PLA
393          TAX
394          PLA
395          TAY
396          PLA
397          PLP
398          RTS
404 ;
405 ASCONV   PHA
406          ASL
407          ASL
408          BCC >1
409          PLA
410          AND #$0F
411          CLC
412          ADC #$9
413          JMP >2
414 ^1       PLA
415          AND #$0F
416 ^2       STA ASCVRES
417          RTS
419 ;
420 ASCVRES  BYT 0
426 ;
427 ASCONV2  LDA INP2CH
428          STA CH
429          LDA INP2CV
430          STA CV
431          JSR VTAB
432          JSR RDKEY
433          JSR COUT
434          JSR ASCONV
435          LDA ASCVRES
436          ASL
437          ASL
438          ASL
439          ASL
440          STA ASCV2RES
441          JSR RDKEY
442          JSR COUT
443          JSR ASCONV
444          CLC
445          LDA ASCVRES
446          ADC ASCV2RES
447          STA ASCV2RES
448          RTS
450 ;
451 ASCV2RES BYT 0
452 INP2CH   BYT 0
453 INP2CV   BYT 0
461 ;
462 OPAFF    LDA OPCH
463          STA CH
464          LDA OPCV
465          STA CV
466          JSR VTAB
467          JSR SETINV
468          LDX OPIND
469          LDA TABLEB,X
470          LDY TABLEH,X
471          JSR STROUT
472          JSR SETNORM
473          RTS
475 ;
476 OPEFF    LDA OPCH
477          STA CH
478          LDA OPCV
479          STA CV
480          JSR VTAB
481          LDX #!20
482          JSR PRBLK2
483          RTS
485 ;
486 OPTEST   LDA KBD
487          BPL OPTEST
488          BIT KBDSTRB
489          CMP #LARROW
490          BNE >1
491          JSR OPEFF
492          DEC OPIND
493          LDA OPIND
494          CMP #$FF
495          BNE >2
496          LDA OPINDMAX
497          STA OPIND
498          DEC OPIND
499 ^2       JSR OPAFF
500          JMP OPTEST
501 ^1       CMP #RARROW
502          BNE >3
503          JSR OPEFF
504          INC OPIND
505          LDA OPIND
506          CMP OPINDMAX
507          BNE >4
508          LDA #0
509          STA OPIND
510 ^4       JSR OPAFF
511          JMP OPTEST
512 ^3       CMP #RETURN
513          BNE >5
514          LDA OPIND
515          CMP #0
516          BNE >1
517          INC FIL1
518          JSR AF.FIL
519          JMP OPTEST
520 ^5       CMP #$A0
521          BNE >6
522          JSR AFF.256
523          INC FIL2
524          JSR AF.FIL
525          JMP OPTEST
526 ^6       CMP #CTRLZ
527          BNE OPTEST
528          JSR PRINT256
529          JMP OPTEST
530 ^1       CMP #1
531          BNE >2
532          DEC FIL1
533          JSR AF.FIL
534          JMP OPTEST
535 ^2       CMP #2
536          BNE >3
537          INC FIL2
538          JSR AF.FIL
539          JMP OPTEST
540 ^3       CMP #3
541          BNE >4
542          DEC FIL2
543          JSR AF.FIL
544          JMP OPTEST
545 ^4       CMP #4
546          BNE >5
547          INC FIL3
548          JSR AF.FIL
549          JMP OPTEST
550 ^5       CMP #5
551          BNE >6
552          DEC FIL3
553          JSR AF.FIL
554          JMP OPTEST
555 ^6       CMP #6
556          BNE >7
557          LDA #AND
558          STA FIL1
559          LDA #$EA
560          STA FIL3
561          JSR AF.FIL
562          JMP OPTEST
563 ^7       CMP #7
564          BNE >8
565          LDA #ORA
566          STA FIL1
567          LDA #$EA
568          STA FIL3
569          JSR AF.FIL
570          JMP OPTEST
571 ^8       CMP #8
572          BNE >9
573          LDA #EOR
574          STA FIL1
575          LDA #$EA
576          STA FIL3
577          JSR AF.FIL
578          JMP OPTEST
579 ^9       CMP #9
580          BNE >1
581          LDA #$20
582          STA FIL1
583          LDA #ECH2BITS
584          STA FIL2
585          LDA /ECH2BITS
586          STA FIL3
587          JSR AF.FIL
588          LDA MEM1
589          STA ECHE1
590          LDA MEM2
591          STA ECHE2
592          JSR AFF.256
593          JMP OPTEST
595 ;
596 ^1       CMP #!10
```

```
597          BNE >2
598          NOP
600 ;
601          JMP ECR2
602 ;
603 ^2       CMP #!11
604          BNE >3
605          JSR AFF.256
606          JMP OPTEST
607 ^3       CMP #!12
608          BNE >4
609          INC MEM1
610          JSR AF.MEM
611          JMP OPTEST
612 ^4       CMP #!13
613          BNE >5
614          DEC MEM1
615          JSR AF.MEM
616          JMP OPTEST
617 ^5       CMP #!14
618          BNE >6
619          INC MEM2
620          JSR AF.MEM
621          JMP OPTEST
622 ^6       CMP #!15
623          BNE >7
624          DEC MEM2
625          JSR AF.MEM
626          JMP OPTEST
627 ^7       CMP #!16
628          BNE >8
629          INC IBTRK
630          LDA IBTRK
631          CMP #$23
632          BNE OPSUIV1
633          LDA #0
634          STA IBTRK
635 OPSUIV1  JSR AFFTRSC
636          JSR RWTS
637          JSR AFF.256
638          JMP OPTEST
639 ^8       CMP #!17
640          BNE OPTXIT
641          INC IBSECT
642          LDA IBSECT
643          CMP #$10
644          BNE OPSUIV2
645          LDA #0
646          STA IBSECT
647 OPSUIV2  JSR AFFTRSC
648          JSR RWTS
649          JSR AFF.256
650          JMP OPTEST
651 OPTXIT   JMP OPTEST
654 ;
655          RTS
658 ;
659 OPCV     BYT !18
660 OPCH     BYT 0
661 OPIND    BYT 0
662 OPINDMAX BYT 16
667 ;
668 C1   ASC "INCREMENTE OCTET 1"
669      HEX 00
670 C2   ASC "DECREMENTE OCTET 1"
671      HEX 00
672 C3   ASC "INCREMENTE OCTET 2"
673      HEX 00
674 C4   ASC "DECREMENTE OCTET 2"
675      HEX 00
676 C5   ASC "INCREMENTE OCTET 3"
677      HEX 00
678 C6   ASC "DECREMENTE OCTET 3"
679      HEX 00
680 C7   ASC "ET LOGIQUE"
681      HEX 00
682 C8   ASC "OU LOGIQUE"
683      HEX 00
684 C9   ASC "OU EXCLUSIF"
685      HEX 00
686 C10  ASC "ECHANGE DE BITS"
687      HEX 00
688 C11  ASC "LIRE SECTEUR"
689      HEX 00
690 C12  ASC "EXECUTION"
691      HEX 00
692 C13  ASC "INCREMENTE M.1"
693      HEX 00
694 C14  ASC "DECREMENTE M.1"
695      HEX 00
696 C15  ASC "INCREMENTE M.2"
697      HEX 00
698 C16  ASC "DECREMENTE M.2"
699      HEX 00
700 C17  ASC "PISTE SUIVANTE"
701      HEX 00
702 C18  ASC "SECTEUR SUIVANT"
703      HEX 00
707 ;
708 TABLEB   BYT C1
709          BYT C2
710          BYT C3
711          BYT C4
712          BYT C5
713          BYT C6
714          BYT C7
715          BYT C8
716          BYT C9
717          BYT C10
718          BYT C11
719          BYT C12
720          BYT C13
721          BYT C14
722          BYT C15
723          BYT C16
724          BYT C17
725          BYT C18
726 ;
727 TABLEH   HBY C1
728          HBY C2
729          HBY C3
730          HBY C4
731          HBY C5
732          HBY C6
733          HBY C7
734          HBY C8
735          HBY C9
736          HBY C10
737          HBY C11
738          HBY C12
739          HBY C13
740          HBY C14
741          HBY C15
742          HBY C16
743          HBY C17
744          HBY C18
752 ;
753 MEM1     BYT 1
754 MEM2     BYT 0
756 ;
757 AF.MEM   LDA #!34
758          STA CH
759          LDA #!22
760          STA CV
761          JSR VTAB
762          LDA MEM1
763          JSR PRBYTE
764          LDA #$A0
765          JSR COUT
766          LDA MEM2
767          JSR PRBYTE
768          RTS
794 ;
795 OCTRES   HEX 00000000
797          HEX 00000000
798 OCT1OCT8:
800 ;
801          STY O1O8SAVY
802          CLC
803          LDY #0
804 ^1       LSR
805          BCS >2
806          PHA
807          LDA #0
808          STA OCTRES,Y
809          PLA
810          JMP >3
811 ^2       PHA
812          LDA #1
813          STA OCTRES,Y
814          PLA
815 ^3       INY
816          CPY #8
817          BNE <1
818          LDY O1O8SAVY
819          RTS
821 ;
822 O1O8SAVY BYT 0
824 ;
825 OCT8OCT1:
827 ;
828          STY O8O1SAVY
829          LDA #0
830          STA O8O1RES
831          LDY #7
832 ^1       LDA OCTRES,Y
833          BNE >2
834          CLC
835          ROL O8O1RES
836          JMP >3
837 ^2       SEC
838          ROL O8O1RES
839 ^3       DEY
840          BPL <1
841          LDY O8O1SAVY
842          RTS
843 ;
844 O8O1RES  BYT 0
845 O8O1SAVY BYT 0
847 ;
848 ECH2BITS:
851 ;
852          STX ECBSAVX
853          JSR OCT1OCT8
854          LDX ECHE1
855          LDA OCTRES,X
856          PHA
857          LDX ECHE2
858          LDA OCTRES,X
859          LDX ECHE1
860          STA OCTRES,X
861          LDX ECHE2
862          PLA
863          STA OCTRES,X
864          JSR OCT8OCT1
865          LDA O8O1RES
866          STA ECHRES
867          LDX ECBSAVX
868          RTS
870 ;
871 ECHE1    BYT 0
872 ECHE2    BYT 0
873 ;
874 ECHRES   BYT 0
875 ECBSAVX  BYT 0
882 ;
883 ALG1     CLC
884          ADC MEM1
885          PHA
886          LDA MEM1
887          BNE >1
888          JSR SOUND0
889 ^1       INC MEM1
890          JSR AF.MEM
891          PLA
892          RTS
895 ;
896 ALG2     ASL
897          PHA
898          JSR SOUND0
899          PLA
900          RTS
903 ;
904 ALG3     LSR
905          PHA
906          JSR SOUND0
907          PLA
908          RTS
911 ;
912 ALG4     ASL
913          ASL
914          PHA
915          JSR SOUND0
916          PLA
917          RTS
920 ;
921 ALG5     LSR
922          LSR
923          PHA
924          JSR SOUND0
925          PLA
926          RTS
929 ;
930 ALG6     ASL
931          CLC
932          ADC MEM1
933          PHA
934          LDA MEM1
935          BNE >1
936          JSR SOUND0
937 ^1       INC MEM1
938          JSR AF.MEM
939          PLA
940          RTS
943 ;
944 ALG7     LSR
945          CLC
946          ADC MEM1
947          PHA
948          LDA MEM1
949          BNE >1
950          JSR SOUND0
951 ^1       INC MEM1
952          JSR AF.MEM
953          RTS
956 ;
957 ALG8     ASL
958          ASL
959          CLC
960          ADC MEM1
961          PHA
962          LDA MEM1
963          BNE >1
964          JSR SOUND0
965 ^1       INC MEM1
966          JSR AF.MEM
967          PLA
968          RTS
971 ;
972 ALG9     LSR
973          LSR
974          CLC
975          ADC MEM1
976          PHA
977          LDA MEM1
978          BNE >1
979          JSR SOUND0
980 ^1       INC MEM1
981          JSR AF.MEM
982          PLA
983          RTS
989 ;
990 SOUND0:
991          JSR SNDSAVRG
992          LDX #$B0
993 ^1       TXA
994          EOR #$80
995          TAY
996 ^2       DEY
997          BNE <2
998          BIT SPKR
999          TXA
1000         ADC #$80
1001         TAY
1002 ^3      DEY
1003         BNE <3
1004         BIT SPKR
1005         DEX
1006         BNE <1
1007         JSR SNDRSTRG
1008         RTS
1011 ;
1012 SOUND1:
1013         JSR SNDSAVRG
1014         LDA SFLAG
1015         BEQ >2
1016         LDA #1
1017         STA SFLAG
1018 ^3      LDX #0
1019 ^1      LDA SPKR,X
1020         INX
1021         CPX #!30
1022         BNE <1
1023         INC SFLAG
1024         LDA SFLAG
1025         CMP #5
1026         BNE <3
1027         JSR SNDRSTRG
1028 ^2      RTS
1031 ;
1032 SFLAG   BYT 1
1033 SNDSAVA BYT 0
1034 SNDSAVX BYT 0
1035 SNDSAVY BYT 0
1038 ;
1039 SNDSAVRG STA SNDSAVA
1040         STX SNDSAVX
1041         STY SNDSAVY
1042         RTS
1045 ;
1046 SNDRSTRG LDA SNDSAVA
1047         LDX SNDSAVX
1048         LDY SNDSAVY
1049         RTS
1056 ;
1057 PRINT256 STA PTSAVA
1058         STY PTSAVY
1059         LDA $400
1060         STA PTSAV400
1061 ;
1062         JSR WNDSAV
1063         JSR WNDINIT
1064 ;
1065         LDA #1
1066         JSR OUTPORT
1067         LDY #0
1068 ^1      LDA (BUF),Y
1069 PFIL1   NOP
1070 PFIL2   NOP
1071 PFIL3   NOP
1072         STY PTSAVY2
1073         ORA #$80
1074         CMP #$FF
1075         BEQ >3
1076         CMP #$80
1077         BLT >2
1078         CMP #$A0
1079         BGE >2
```

```
1080 ^3        LDA #$AE
1081 ^2        JSR COUTLF
1082           LDY PTSAVY2
1083           INY
1084           CPY #0
1085           BNE <1
1086           LDA #$8D
1087           JSR COUTLF
1088           LDA #0
1089           JSR OUTPORT
1090 ;
1091           JSR WNDRST
1092 ;
1093           LDA PTSAV400
1094           LDA PTSAVA
1095           LDY PTSAVY
1096           RTS
1099 ;
1100 PTSAVA    BYT 0
1101 PTSAVY    BYT 0
1102 PTSAV400  BYT 0
1103 PTSAVY2   BYT 0
1106 ;
1107 WNDINIT   LDA #0
1108           STA WNDLFT
1109           STA WNDTOP
1110           STA CH
1111           STA CV
1112           JSR VTAB
1113           LDA #1
1114           STA WNDWDTH
1115           STA WNDBTM
1116           RTS
1117 ;
1118 WNDSAV    LDA WNDLFT
1119           STA WLFSAV
1120           LDA WNDTOP
1121           STA WTOPSAV
1122           LDA WNDWDTH
1123           STA WWDTHSAV
1124           LDA WNDBTM
1125           STA WBTMSAV
1126           LDA CH
1127           STA WCHSAV
1128           LDA CV
1129           STA WCVSAV
1130           RTS
1131 ;
1132 WNDRST    LDA WLFSAV
1133           STA WNDLFT
1134           LDA WTOPSAV
1135           STA WNDTOP
1136           LDA WWDTHSAV
1137           STA WNDWDTH
1138           LDA WBTMSAV
1139           STA WNDBTM
1140           LDA WCHSAV
1141           STA CH
1142           LDA WCVSAV
1143           STA CV
1144           JSR VTAB
1145           RTS
1146 ;
1147 WLFSAV    BYT 0
1148 WTOPSAV   BYT 0
1149 WWDTHSAV  BYT 0
1150 WBTMSAV   BYT 0
1151 WCHSAV    BYT 0
1152 WCVSAV    BYT 0
1157 ;
1158 INITVECT  LDA #INTRO
1159           STA $3F2
1160           LDA /INTRO
1161           STA $3F3
1163 ;
1164           LDA $3F3
1165           EOR #$A5
1166           STA $3F4
1167           RTS
1171 ;
1172 COUTLF    PHA
1173           LDA COMPTLF
1174           CMP #!60
1175           BNE >1
1176           LDA #0
1177           STA COMPTLF
1178           STA CH
1179           LDA #$8A
1180           JSR COUT
1181 ^1        INC COMPTLF
1182           PLA
1183           JSR COUT
1184           RTS
1187 ;
1188 COMPTLF   BYT 0
1191 ;
1192 STROUTLF  STA STRADR
1193           STY STRADR+$1
1194           LDY #0
1195 ^1        LDA (STRADR),Y
1196           BEQ >2
1197           JSR COUTLF
1198           INY
1199           CLC
1200           BCC <1
1201 ^2        RTS
1204 ;
1205 STRADR    EPZ $EB
1206 LFFLAG    BYT 1
1209 ;
1210 LFLF      LDA LFFLAG
1211           BEQ >1
1212           LDA #$8A
1213           JSR COUT
1214           LDA #0
1215           STA CH
1216           STA COMPTLF
1217 ^1        RTS
1220 ;
1221           END
```

Récapitulation DECRYPTEUR

```
0800- 20 39 12 20 58 FC 20 68
0808- 0C 8D 8D 8D D0 D2 CF C7
0810- D2 C1 CD CD C5 A0 C4 C5
0818- A0 C4 C5 C3 D2 D9 D0 D4
0820- C1 C7 C5 8D AD AD AD AD
0828- AD AD AD AD AD AD AD AD
0830- AD AD AD AD AD AD AD AD
0838- AD AD AD 8D C4 C5 A0 D3
0840- C5 C3 D4 C5 D5 D2 D3 A0
0848- A0 C3 CF C4 C5 D3 A0 D0
0850- C1 D2 A0 8D D5 D4 C9 CC
0858- C9 D3 C1 D4 C9 CF CE A0
0860- C4 A7 D5 CE A0 C6 C9 CC
0868- D4 D2 C5 8D AD AD AD AD
0870- AD AD AD AD AD AD AD AD
0878- AD AD AD AD AD AD AD AD
0880- AD AD AD 8D 8D 8D A0 A0
0888- A0 A0 A0 D0 C1 D2 A0 D0
0890- C1 D5 CC A0 C6 D2 C1 CE
0898- C3 C5 D3 C3 C8 C9 8D A0
08A0- A0 A0 A0 A0 C3 CF D0 D9
08A8- D2 C9 C7 C8 D4 A0 B1 B9
08B0- B8 B5 AE 16 05 12 13 09
08B8- 0F 0E 20 31 2E 31 00 AD
08C0- 00 C0 10 FB 2C 10 C0 20
08C8- 58 FC 20 68 0C C3 C5 A0
08D0- D0 D2 CF C7 D2 C1 CD CD
08D8- C5 A0 D6 CF D5 D3 A0 D0
08E0- C5 D2 CD C5 D4 A0 C4 A7
08E8- C1 C6 C6 C9 C3 C8 C5 D2
08F0- A0 CC C5 8D C3 CF CE D4
08F8- C5 CE D5 A0 C4 A7 D5 CE
0900- A0 D3 C5 C3 D4 C5 D5 D2
0908- A0 C4 C5 A0 CD D5 CC D4
0910- C9 D0 CC C5 D3 A0 A0 C6
0918- C1 AD 8D C3 CF CE D3 AC
0920- A0 C5 CE A0 C3 CF CD CD
0928- D5 D4 C1 CE D4 A0 C1 A0
0930- A0 C3 C8 C1 D1 D5 C5 A0
0938- A0 C6 CF C9 D3 A0 A0 D5
0940- CE 8D C6 C9 CC D4 D2 C5
0948- AC A0 CD C1 D4 C5 D2 C9
0950- C1 CC C9 D3 C5 A0 D0 C1
0958- D2 A0 D4 D2 CF C9 D3 A0
0960- CF C3 D4 C5 D4 D3 AE 8D
0968- 8D D6 CF D5 D3 A0 D0 CF
0970- D5 D6 C5 DA A0 C1 C9 CE
0978- D3 C9 A0 D6 C9 D3 D5 C1
0980- CC C9 D3 C5 D2 A0 D5 CE
0988- A0 A0 D3 C5 C3 D4 AD 8D
0990- C5 D5 D2 A0 C1 D6 C5 C3
0998- A0 D5 CE A0 C6 C9 CC D4
09A0- D2 C5 A0 C3 CC C1 C9 D2
09A8- AC A0 C6 CF CE C3 C5 AC
09B0- A0 A0 CF D0 C1 AD 8D D1
09B8- D5 C5 A0 CF D5 A0 CF D2
09C0- C1 CE C7 C5 BB A0 D6 CF
09C8- D4 D2 C5 A0 D6 C9 D3 C9
09D0- CF CE A0 D0 C5 D5 D4 A0
09D8- A0 C5 D4 D2 C5 8D C6 CC
09E0- CF D5 C5 AC A0 D0 D2 C5
09E8- C3 C9 D3 C5 AC A0 C4 C5
09F0- A0 C8 C1 D5 D4 AC A0 C4
09F8- C5 A0 D0 D2 C5 D3 A0 AC
0A00- A0 C4 C5 A0 8D C2 C9 C1
0A08- C9 D3 AC A0 CF D5 A0 C5
0A10- D8 D4 D2 C1 CC D5 C3 C9
0A18- C4 C5 AE AE AE 8D 8D 8D
0A20- C1 CC CF D2 D3 AC A0 D1
0A28- D5 C5 CC A0 D3 C5 C3 D4
0A30- C5 D5 D2 A0 C4 C5 D3 C9
0A38- D2 C5 DA AD D6 CF D5 D3
0A40- A0 CC C9 D2 C5 BF 8D 8D
0A48- 8D 20 20 10 09 13 14 05
0A50- 20 20 20 20 20 8D 8D 20
0A58- 20 13 05 03 14 05 15 12
0A60- 20 20 20 00 A9 0E 8D EC
0A68- 0C A9 0F 8D ED 0C 20 B7
0A70- 0C AD EB 0C 8D 2F 0C A9
0A78- 0E 8D EC 0C EE ED 0C EE
0A80- ED 0C 20 B7 0C AD EB 0C
0A88- 8D 30 0C A9 0A 85 24 A9
0A90- 16 85 25 20 22 FC 20 68
0A98- 0C 45 53 50 41 43 45 60
0AA0- 50 4F 55 52 60 43 4F 4E
0AA8- 54 49 4E 55 45 52 00 AD
0AB0- 00 C0 10 FB 2C 10 C0 20
0AB8- 58 FC 20 68 0C A0 A0 A0
0AC0- A0 A0 A0 A0 A0 AA AA AA
0AC8- AA AA AA AA AA AA AA AA
0AD0- AA AA AA AA AA AA AA 8D
0AD8- A0 A0 A0 A0 A0 A0 A0 A0
0AE0- AA A0 C4 C5 C3 D2 D9 D0
0AE8- D4 C5 D5 D2 A0 B1 AE B0
0AF0- A0 AA 8D A0 A0 A0 A0 A0
0AF8- A0 A0 A0 AA AA AA AA AA
0B00- AA AA AA AA AA AA AA AA
0B08- AA AA AA AA AA 00 20 14
0B10- 0B 4C 44 0B A9 04 85 25
0B18- 20 22 FC A9 00 85 24 20
0B20- 68 0C D0 C9 D3 D4 C5 A0
0B28- 00 AD 2F 0C 20 DA FD 20
0B30- 68 0C A0 A0 D3 C5 C3 D4
0B38- C5 D5 D2 A0 00 AD 30 0C
0B40- 20 DA FD 60 A9 00 8D 14
0B48- 0C 85 06 A9 60 85 07 20
0B50- 19 0C 20 BE 0B A9 21 85
0B58- 24 A9 02 85 25 20 22 FC
0B60- 20 68 0C C6 C9 CC D4 D2
0B68- C5 00 A9 21 85 24 A9 03
0B70- 85 25 20 22 FC 20 68 0C
0B78- AD AD AD AD AD AD 00 A9
0B80- EA 8D D1 0B 8D D2 0B 8D
0B88- D3 0B 20 40 0C A9 21 85
0B90- 24 A9 14 85 25 20 22 FC
0B98- 20 68 0C CD AE B1 A0 CD
0BA0- AE B2 00 20 01 10 A9 00
0BA8- 8D C3 0E 8D C2 0E A9 14
0BB0- 8D C1 0E A9 12 8D C4 0E
```

```
0BB8- 20 EE 0C 4C 21 0D A9 00
0BC0- 85 24 8D 14 0C A9 08 85
0BC8- 25 20 22 FC A0 00 B1 06
0BD0- EA EA EA EA C9 80 90 06
0BD8- C9 A0 B0 02 A9 AE 20 ED
0BE0- FD C8 EE 14 0C AD 14 0C
0BE8- C9 20 D0 E2 A9 00 8D 14
0BF0- 0C A9 00 85 24 E6 25 20
0BF8- 22 FC C0 00 D0 D0 20 4A
0C00- 11 AD D1 0B 8D A2 11 AD
0C08- D2 0B 8D A3 11 AD D3 0B
0C10- 8D A4 11 60 00 00 00 00
0C18- 00 A9 3C 8D 31 0C A9 0C
0C20- 8D 32 0C A9 0C A0 2B 20
0C28- D9 03 60 01 60 01 00 00
0C30- 00 00 00 00 60 00 00 01
0C38- 00 00 60 01 00 01 EF D8
0C40- A9 05 85 25 20 22 FC A9
0C48- 20 85 24 AD D1 0B 20 DA
0C50- FD A9 A0 20 ED FD AD D2
0C58- 0B 20 DA FD A9 A0 20 ED
0C60- FD AD D3 0B 20 DA FD 60
0C68- 08 48 98 48 8A 48 BA BD
0C70- 05 01 85 08 BD 06 01 85
0C78- 09 A0 01 B1 08 F0 0B 20
0C80- ED FD E6 08 D0 F5 E6 09
0C88- D0 F1 E6 08 D0 02 E6 09
0C90- A5 08 9D 05 01 A5 09 9D
0C98- 06 01 68 AA 68 A8 68 28
0CA0- 60 48 0A 0A 90 09 68 29
0CA8- 0F 18 69 09 4C B2 0C 68
0CB0- 29 0F 8D B6 0C 60 00 AD
0CB8- EC 0C 85 24 AD ED 0C 85
0CC0- 25 20 22 FC 20 0C FD 20
0CC8- ED FD 20 A1 0C AD B6 0C
0CD0- 0A 0A 0A 0A 8D EB 0C 20
0CD8- 0C FD 20 ED FD 20 A1 0C
0CE0- 18 AD B6 0C 6D EB 0C 8D
0CE8- EB 0C 60 00 00 00 AD C2
0CF0- 0E 85 24 AD C1 0E 85 25
0CF8- 20 22 FC 20 80 FE AE C3
0D00- 0E BD DB 0F BC ED 0F 20
0D08- 3A DB 20 84 FE 60 AD C2
0D10- 0E 85 24 AD C1 0E 85 25
0D18- 20 22 FC A2 14 20 4A F9
0D20- 60 AD 00 C0 10 FB 2C 10
0D28- C0 C9 88 D0 1C 20 0E 0D
0D30- CE C3 0E AD C3 0E C9 FF
0D38- D0 09 AD C4 0E 8D C3 0E
0D40- CE C3 0E 20 EE 0C 4C 21
0D48- 0D C9 95 D0 19 20 0E 0D
0D50- EE C3 0E AD C3 0E CD C4
0D58- 0E D0 05 A9 00 8D C3 0E
0D60- 20 EE 0C 4C 21 0D C9 8D
0D68- D0 10 AD C3 0E C9 00 D0
0D70- 23 EE D1 0B 20 40 0C 4C
0D78- 21 0D C9 A0 D0 0C 20 BE
0D80- 0B EE D2 0B 20 40 0C 4C
0D88- 21 0D C9 9A D0 93 20 87
0D90- 11 4C 21 0D C9 01 D0 09
0D98- CE D1 0B 20 40 0C 4C 21
0DA0- 0D C9 02 D0 09 EE D2 0B
0DA8- 20 40 0C 4C 21 0D C9 03
0DB0- D0 09 CE D2 0B 20 40 0C
0DB8- 4C 21 0D C9 04 D0 09 EE
0DC0- D3 0B 20 40 0C 4C 21 0D
0DC8- C9 05 D0 09 CE D3 0B 20
0DD0- 40 0C 4C 21 0D C9 06 D0
0DD8- 10 A9 29 8D D1 0B A9 EA
0DE0- 8D D3 0B 20 40 0C 4C 21
0DE8- 0D C9 07 D0 10 A9 09 8D
0DF0- D1 0B A9 EA 8D D3 0B 20
0DF8- 40 0C 4C 21 0D C9 08 D0
0E00- 10 A9 49 8D D1 0B A9 EA
0E08- 8D D3 0B 20 40 0C 4C 21
0E10- 0D C9 09 D0 24 A9 20 8D
0E18- D1 0B A9 6D 8D D2 0B A9
0E20- 10 8D D3 0B 20 40 0C AD
0E28- FF 0F 8D 9A 10 AD 00 10
0E30- 8D 9B 10 20 BE 0B 4C 21
0E38- 0D C9 0A D0 04 EA 4C C7
0E40- 08 C9 0B D0 06 20 BE 0B
0E48- 4C 21 0D C9 0C D0 09 EE
0E50- FF 0F 20 01 10 4C 21 0D
0E58- C9 0D D0 09 CE FF 0F 20
0E60- 01 10 4C 21 0D C9 0E D0
0E68- 09 EE 00 10 20 01 10 4C
0E70- 21 0D C9 0F D0 09 CE 00
0E78- 10 20 01 10 4C 21 0D C9
0E80- 10 D0 1B EE 2F 0C AD 2F
0E88- 0C C9 23 D0 05 A9 00 8D
0E90- 2F 0C 20 14 0B 20 19 0C
0E98- 20 BE 0B 4C 21 0D C9 11
0EA0- D0 1B EE 30 0C AD 30 0C
0EA8- C9 10 D0 05 A9 00 8D 30
0EB0- 0C 20 14 0B 20 19 0C 20
0EB8- BE 0B 4C 21 0D 4C 21 0D
0EC0- 60 12 00 00 10 C9 CE C3
0EC8- D2 C5 CD C5 CE D4 C5 A0
0ED0- CF C3 D4 C5 D4 A0 B1 00
0ED8- C4 C5 C3 D2 C5 CD C5 CE
0EE0- D4 C5 A0 CF C3 D4 C5 D4
0EE8- A0 B1 00 C9 CE C3 D2 C5
0EF0- CD C5 CE D4 C5 A0 CF C3
0EF8- D4 C5 D4 A0 B2 00 C4 C5
0F00- C3 D2 C5 CD C5 CE D4 C5
0F08- A0 CF C3 D4 C5 D4 A0 B2
0F10- 00 C9 CE C3 D2 C5 CD C5
0F18- CE D4 C5 A0 CF C3 D4 C5
0F20- D4 A0 B3 00 C4 C5 C3 D2
0F28- C5 CD C5 CE D4 C5 A0 CF
0F30- C3 D4 C5 D4 A0 B3 00 C5
0F38- D4 A0 CC CF C7 C9 D1 D5
0F40- C5 00 CF D5 A0 CC CF C7
0F48- C9 D1 D5 C5 00 CF D5 A0
0F50- C5 D8 C3 CC D5 D3 C9 C6
0F58- 00 C5 C3 C8 C1 CE C7 C5
0F60- A0 C4 C5 A0 C2 C9 D4 D3
0F68- 00 CC C9 D2 C5 A0 D3 C5
0F70- C3 D4 C5 D5 D2 00 C5 D8
0F78- C5 C3 D5 D4 C9 CF CE 00
0F80- C9 CE C3 D2 C5 CD C5 CE
0F88- D4 C5 A0 CD AE B1 00 C4
0F90- C5 C3 D2 C5 CD C5 CE D4
0F98- C5 A0 CD AE B1 00 C9 CE
0FA0- C3 D2 C5 CD C5 CE D4 C5
0FA8- A0 CD AE B2 00 C4 C5 C3
0FB0- D2 C5 CD C5 CE D4 C5 A0
0FB8- CD AE B2 00 D0 C9 D3 D4
0FC0- C5 A0 D3 D5 C9 D6 C1 CE
0FC8- D4 C5 00 D3 C5 C3 D4 C5
0FD0- D5 D2 A0 D3 D5 C9 D6 C1
0FD8- CE D4 00 C5 D8 EB FE 11
0FE0- 24 37 42 4D 59 69 76 80
0FE8- 8F 9E AD BC CB 0E 0E 0E
0FF0- 0E 0F 0F 0F 0F 0F 0F 0F
0FF8- 0F 0F 0F 0F 0F 0F 0F 01
1000- 00 A9 22 85 24 A9 16 85
1008- 25 20 22 FC AD FF 0F 20
1010- DA FD A9 A0 20 ED FD AD
1018- 00 10 20 DA FD 60 00 00
1020- 00 00 00 00 00 00 8C 49
1028- 10 18 A0 00 4A B0 0A 48
1030- A9 00 99 1E 10 68 4C 40
1038- 10 48 A9 01 99 1E 10 68
1040- C8 C0 08 D0 E7 AC 49 10
1048- 60 00 8C 6C 10 A9 00 8D
1050- 6B 10 A0 07 B9 1E 10 D0
1058- 07 18 2E 6B 10 4C 64 10
1060- 38 2E 6B 10 88 10 ED AC
1068- 6C 10 60 00 00 8E 9D 10
1070- 20 26 10 AE 9A 10 BD 1E
1078- 10 48 AE 9B 10 BD 1E 10
1080- AE 9A 10 9D 1E 10 AE 9B
1088- 10 68 9D 1E 10 20 4A 10
1090- AD 6B 10 8D 9C 10 AE 9D
1098- 10 60 00 00 00 00 18 6D
10A0- FF 0F 48 AD FF 0F D0 03
10A8- 20 2A 11 EE FF 0F 20 01
10B0- 10 68 60 0A 48 20 2A 11
10B8- 68 60 4A 48 20 2A 11 68
10C0- 60 0A 0A 48 20 2A 11 68
10C8- 60 4A 4A 48 20 2A 11 68
10D0- 60 0A 18 6D FF 0F 48 AD
10D8- FF 0F D0 03 20 2A 11 EE
10E0- FF 0F 20 01 10 68 60 4A
10E8- 18 6D FF 0F 48 AD FF 0F
10F0- D0 03 20 2A 11 EE FF 0F
10F8- 20 01 10 60 0A 0A 18 6D
1100- FF 0F 48 AD FF 0F D0 03
1108- 20 2A 11 EE FF 0F 20 01
1110- 10 68 60 4A 4A 18 6D FF
1118- 0F 48 AD FF 0F D0 03 20
1120- 2A 11 EE FF 0F 20 01 10
1128- 68 60 20 73 11 A2 B0 8A
1130- 49 80 A8 88 D0 FD 2C 30
1138- C0 8A 69 80 A8 88 D0 FD
1140- 2C 30 C0 CA D0 E9 20 7D
1148- 11 60 20 73 11 AD 6F 11
1150- F0 1C A9 01 8D 6F 11 A2
1158- 00 BD 30 C0 E8 E0 1E D0
1160- F8 EE 6F 11 AD 6F 11 C9
1168- 05 D0 EC 20 7D 11 60 01
1170- 00 00 00 8D 70 11 8E 71
1178- 11 8C 72 11 60 AD 70 11
1180- AE 71 11 AC 72 11 60 8D
1188- DA 11 8C DB 11 AD 00 04
1190- 8D DC 11 20 F2 11 20 DE
1198- 11 A9 01 20 95 FE A0 00
11A0- B1 06 EA EA EA 8C DD 11
11A8- 09 80 C9 FF F0 08 C9 80
11B0- 90 06 C9 A0 B0 02 A9 AE
11B8- 20 4C 12 AC DD 11 C8 C0
11C0- 00 D0 DD A9 8D 20 4C 12
11C8- A9 00 20 95 FE 20 11 12
11D0- AD DC 11 AD DA 11 AC DB
11D8- 11 60 00 00 00 00 A9 00
11E0- 85 20 85 22 85 24 85 25
11E8- 20 22 FC A9 01 85 21 85
11F0- 23 60 A5 20 8D 33 12 A5
11F8- 22 8D 34 12 A5 21 8D 35
1200- 12 A5 23 8D 36 12 A5 24
1208- 8D 37 12 A5 25 8D 38 12
1210- 60 AD 33 12 85 20 AD 34
1218- 12 85 22 AD 35 12 85 21
1220- AD 36 12 85 23 AD 37 12
1228- 85 24 AD 38 12 85 25 20
1230- 22 FC 60 00 00 00 00 00
1238- 00 A9 03 8D F2 03 A9 08
1240- 8D F3 03 AD F3 03 49 A5
1248- 8D F4 03 60 48 AD 68 12
1250- C9 3C D0 0C A9 00 8D 68
1258- 12 85 24 A9 8A 20 ED FD
1260- EE 68 12 68 20 ED FD 60
1268- 00 85 EB 84 EC A0 00 B1
1270- EB F0 07 20 4C 12 C8 18
1278- 90 F5 60 01 AD 7B 12 F0
1280- 0C A9 8A 20 ED FD A9 00
1288- 85 24 8D 68 12 60
```

Nous avons les moye

C.L.M.& BBDO Apple et le logo Apple sont des marques déposées d'Apple Computer, Inc. Mac Tell 2 (Copyright 1985 - Hello Informatique) matériel agréé par les P.T.T.

Modem 300 bauds.

Autant l'avouer tout de suite, nous sommes dans l'ère de l'information. Et avant de pouvoir analyser les données qui proviennent des quatre coins du monde, il faut déjà pouvoir les attraper.

Vous pouvez toujours chauffer la plante des pieds de vos employés, sans un Macintosh, un bon modem et un programme de communication, impossible de leur faire dire quoi que ce soit à propos de votre correspondant de Chicago, ni sur la vente de maillots de bain pendant l'été 1985 au pôle Nord. Ne vous torturez pas davantage, nous sommes enfin disposés à tout vous révéler à propos de nos moyens de communication.

Les trois modem Apple sont :
– Le modem 300 bauds, d'usage très simple ; peu cher, il permet à n'importe quel étudiant de chercher partout des renseignements pour son interrogation du lendemain. – Le modem universel 1200/75 Vidéotex et 300 bauds qui permet de communiquer en vidéotext sur des applications pseudo-graphiques (en mode 1200/75) et en texte de façon très ordinaire (en mode 300). Universel, il est idéal pour faire parler les avocats, les médecins ou les dentistes.
– Le modem 1200 bauds full duplex : il fait parler les gros systèmes et ce, à grande vitesse. Lors des échanges de fichiers contenant une quantité importante d'informations, ce modem est une merveille, compte

Apple Computer France. Avenue de l'Océan

ns de vous faire parler.

Modem Full duplex 1200 bauds.

Modem Universel 1200/75 Vidéotex 300 bauds.

tenu de l'existence de "protocoles", qui sur des liaisons spécialisées 1200 bauds, se chargent de la vérification de la validité des informations transmises, évitant ainsi la plupart des erreurs de transmission. C'est amusant de voir comment une si petite machine peut faire parler si facilement de gigantesques systèmes de données.

Dire que tant d'hommes ont cherché des moyens très compliqués pour obtenir des renseignements, alors qu'un programme de communication comme Mac Tell 2 de Hello Informatique ou Mac Terminal peut faire avouer à n'importe quel serveur d'information, n'importe quoi.

Mac Terminal peut interroger un réseau télématique comme Calvados, messagerie électronique française qui peut vous faire rencontrer beaucoup plus de monde que sur la place de l'Opéra aux heures de pointe. Vous pouvez échanger des informations, connaître les cours de la Bourse, consulter les dépêches de l'AFP. Mac Tell 2 permet d'émuler un Minitel avec tous les avantages d'un Macintosh (mémorisation, impression).

Maintenant, décidez-vous, décidez en étant au courant de tout, ou décidez en étant au courant de rien.

Apple

Z.A. de Courtabœuf, B.P. 131, 91944 Les Ulis Cedex.

Librairie : Un 'motif' de satisfaction

Marianne Sutz

Nous vous proposions, dans le numéro 18 de "Pom's", deux routines qui vous permettaient de récupérer les icônes et les curseurs du système. Cette fois, notre attention va se porter sur les 38 "motifs" (pattern) prédéfinis, utilisés par le tableau de bord pour déterminer l'apparence du "bureau". La définition de motifs dans un programme Basic étant longue et fastidieuse, une telle routine nous semblait indispensable. Ceci va aussi nous permettre de décrire, dans ses grandes lignes, le fonctionnement de l'instruction "LIBRARY" qui n'est évoquée, dans le manuel d'utilisation du Basic Microsoft, que dans la liste des mots-clé. Vous trouverez donc, jointes à cet article, deux versions de la routine :

- une version que nous appelerons "tableau", puisqu'elle se loge dans un tableau de variables entières (c'est la méthode utilisée dans les précédents numéros de "Pom's");
- une version "LIBRARY" qui, comme nous le verrons un peu plus loin, requiert la présence d'un fichier ressource.

Version "Tableau"

La routine doit être implantée dans un tableau de variables entières de vingt éléments, et son adresse de base correspond au premier de ces éléments. Pour obtenir cette adresse (pour le CALL) il faut donc utiliser :

Adr!=VARPTR(Tableau%(0))
avec OPTION BASE 0 (valeur par défaut), ou :

Adr!=VARPTR(Tableau%(1))
avec OPTION BASE 1.

Il suffit ensuite d'appeler la routine en lui passant, comme argument, le numéro de référence du motif désiré. Cette valeur doit être comprise entre 0 et 37 inclus (voir le tableau ci-joint). Un nombre inférieur à 0 ou supérieur à 37 ne provoque pas de "plantage", mais on obtient alors n'importe quoi comme motif. Le numéro peut être passé sous la forme d'une expression ou d'une variable qui doit absolument être entière (2 octets soit 16 bits). Ainsi :

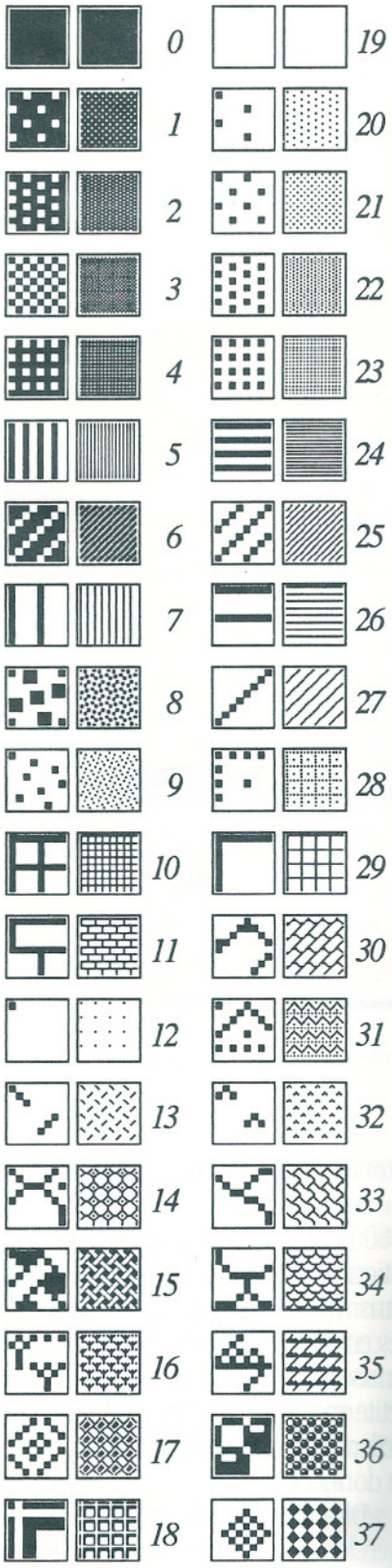
Adr! N%*2+1
et
Adr! N1%
constituent des appels valides de la routine.

Programme de démonstration

```
DEFINT A-Z:DIM R(3),C(19)
DATA &h4E56,0,&h594F,&h2F3C,&h5041
DATA &h5423,&h3F3C,0,&hA9A0,&h225F
DATA &h2251,&h302E,8,&hC0FC,8
DATA &h4871,&h802,&hA89D,&h4E5E
DATA &h4E75
FOR I=0 TO 19:READ C(i):NEXT
FOR I=23 TO 479 STEP 24
LINE(I,0)-(I,300):NEXT
FOR I=23 TO 287 STEP 24
LINE(0,I)-(500,I):NEXT
10 RANDOMIZE TIMER
R(0)=24*INT(RND*13)
R(1)=24*INT(RND*21)
R(2)=R(0)+23:R(3)=R(1)+23
N=INT(RND*38):A!=VARPTR(C(0)):A! N
PAINTRECT VARPTR(R(0)):GOTO 10
```

Version "LIBRARY"

L'instruction "LIBRARY" permet l'utilisation directe de routines créées avec le système de développement 68000, et autorise des liens beaucoup plus étroits entre ces routines et le Basic Microsoft (version 2.00, (d) ou (b)). Sans trop entrer dans les détails (on manque un peu de place !), il convient de citer la



posibilité de générer les messages d'erreur du Basic depuis la routine en assembleur, de contrôler le type des arguments passés, d'utiliser comme arguments des variables qui n'ont pas été préalablement définies, etc... Nous aurons bien sûr l'occasion de reparler de cela dans les prochains numéros de "Pom's" mais, en attendant, nous vous conseillons de demander à la société Microsoft un document polycopié de 21 pages intitulé "Microsoft Basic for the Macintosh. Building Machine Language Libraries", où l'on trouve toutes les adresses liées au système, ainsi que des exemples d'application.

Voyons maintenant quelle est la méthode pour employer la version "LIBRARY" de notre routine.

Elle doit se trouver dans un fichier ressource, qui doit lui-même être situé, de préférence, sur la même disquette que le Basic Microsoft.
Si vous disposez de la disquette "Mac Pom's 21", pas de problème : le fichier s'y trouve (sous le nom "PatLib.Rscr"), de même que les fichiers nécessaires à sa création (PatLib.Job, LIBinit.Asm, PatLib.Asm, PatLib.Link et PatLib.R).
Si vous ne disposez pas de la disquette mais que vous êtes en possession du système de développement 68000, il vous faudra saisir les cinq fichiers précités, puis utiliser le fichier PatLib.Job (avec "Executive"), qui se chargera de l'enchaînement des opérations.
Dans le cas où la disquette "Pom's" et le système de développement 68000 vous feraient défaut, désolé... il n'y a pas de solution. Ceci constitue d'ailleurs le principal inconvénient des routines "LIBRARY" : leur création est impossible sans le système de développement, et il est alors nécessaire de se reporter à la bonne vieille méthode du tableau de variables.

Liaison avec le programme Basic

Avant le premier appel de la routine, nous devons la lier au Basic avec l'instruction "LIBRARY", suivie du nom de la ressource (le fichier contenant la routine en assembleur) ou d'une variable alphanumérique dans laquelle le nom aura été préalablement placé. Dans le programme de démonstration :

```
LIBRARY "PatLib.Rscr"
```

se charge de cette opération.
Si la routine ne se trouve pas sur la même disquette que le Basic Microsoft, il est indispensable que le nom de la ressource soit précédé du nom de volume ("Pom's 21:PatLib.Rscr", par exemple) sinon, on se retrouve avec le message "FILE NOT FOUND". Si le nom "PatLib.Rscr" ne vous convient pas, il est possible de renommer le fichier.

Appel de la routine

On procède de la même façon qu'avec la version "tableau", à une différence près : il n'est pas nécessaire de définir la variable du CALL avec VARPTR et le nom de la routine est imposé par son auteur (nous avons baptisée celle-ci "PatLib"). Notons aussi un petit "plus" par rapport à la version "tableau" : la routine génère le message d'erreur OVERFLOW du Basic si l'argument passé est supérieur à 37 ou inférieur à 0 (bien que, dans ce cas, il s'agirait plutôt de "underflow" mais, ce message n'étant pas disponible...).

Fermeture de la routine

En fait, la fermeture n'est pas indispensable, puisque les instructions NEW, SYSTEM et RUN détachent la routine du Basic. Si, pour une raison quelconque, on doit fermer la routine, on peut utiliser l'instruction LIBRARY CLOSE.

Avant de conclure, il convient de souligner le caractère le plus intéressant du "système LIBRARY" : il permet "l'overlay" dynamique. En effet, si nous passons la valeur 0 dans le

Source de la routine de chargement des "motifs"

```
0000                .Trap  _GetResource     $A9A0    ; Retourne le "Handle" d'une ressource.
0000                .Trap  _PenPat          $A89D    ; Initialise le "motif" à utiliser.
0000
0000   4E56 0000            LINK      A6,#0
0004   594F                 SUBQ      #4,SP          ; 4 octets au sommet de la pile pour "Handle".
0006   2F3C 5041 5423       MOVE.L    #'PAT#',-(SP)  ; Empile le type de ressource à rechercher.
000C   3F3C 0000            MOVE      #0,-(SP)       ; Empile l'identificateur de la ressource (0).
0010   A9A0                 _GetResource             ; Retourne l'adresse du pointeur ("Handle").
0012   225F                 MOVE.L    (SP)+,A1       ; Le "Handle" est placé dans le registre A1.
0014   2251                 MOVEA.L   (A1),A1        ; Le pointeur est placé dans le registre A1.
0016   302E 0008            MOVE      8(A6),D0       ; Le numéro passé depuis le Basic dans D0.
001A   C0FC 0008            MULU      #8,D0          ; Multiplie ce numéro * 8 (1 motif= 8 octets).
001E   4871 08 02           PEA       2(A1,D0.L)     ; Empile l'adresse du motif sélectionné.
0022   A89D                 _PenPat                  ; Change le "motif" du "crayon".
0024   4E5E                 UNLK      A6
0026   4E75                 RTS
```

registre de données D0 avant le retour au Basic, la routine est "purgeable". De ce fait, si la place en mémoire devient insuffisante, la routine sera automatiquement évacuée. Par la suite, un appel de la routine provoquera un rechargement automatique. Ceci s'avère particulièrement utile pour les routines volumineuses et, plus généralement, pour les possesseurs de Mac 128Ko.

Programme de démonstration

Version "LIBRARY"

```
LIBRARY "PatLib.Rscr"
DEFINT A-Z:DIM R(3)
FOR I=23 TO 479 STEP 24
LINE(I,0)-(I,300):NEXT
FOR I=23 TO 287 STEP 24
LINE(0,I)-(500,I):NEXT
10 RANDOMIZE TIMER
R(0)=24*INT(RND*13)
R(1)=24*INT(RND*21)
R(2)=R(0)+23:R(3)=R(1)+23
N=INT(RND*38):PatLib! N
PAINTRECT VARPTR(R(0)):GOTO 10
```

PatLib.Asm

```
.Trap      GetResource     $A9A0
.Trap      _PenPat         $A89D

GetNextLibArg      EQU     $2A
BasicError         EQU     $42

        SUBQ       #4,SP
        MOVE.L     #'PAT#',-(SP)
        MOVE       #0,-(SP)
         GetResource
        MOVE.L     (SP)+,A1
        MOVEA.L    (A1),A1
        JSR        GetNextLibArg(A5)
        MOVE       (A2),D0
        TST        D0
        BMI.S      Erreur
        CMP        #37,D0
        BGT        Erreur
        MULU       #8,D0
        PEA        2(A1,D0.L)
         PenPat
        MOVEQ      #0,D0
        RTS
Erreur  MOVEQ      #6,D2
        JSR        BasicError(A5)
```

PatLib.Link

```
/ OUTPUT PatLib
LIBinit.Rel
<
PatLib.Rel
$
```

MacPaintAstuces

- En mode Crayon et avec le bouton de la souris enfoncé, le déplacement s'obtient avec la touche Option, et le pointage de la loupe à l'endroit de la flèche avec la touche Commande.
- Tous les outils, même le lasso, bénéficient de la direction forcée grâce la touche Majuscules.
- Avec Majuscules, Commande et Option, on peut déformer un objet en préservant sa structure. Ainsi, le cercle de gauche est-il sélectionné par le rectangle; en appuyant sur les trois touches et en tirant vers la droite, on obtient le résultat de la figure 2.

Figure 1

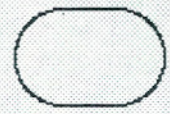


Figure 2

- Si l'on a aussi sélectionné la grille, on préserve non seulement la structure, mais aussi tout décor qui y serait présent (fig. 3).

Figure 3

- Quand on fait des copies multiples, avec les touches Commande et Option, leur espacement est défini par l'épaisseur de trait sélectionnée en bas et à gauche de l'écran.

PatLib.Job

(Executive)

Asm	LIBinit.Asm	Exec	Edit
Asm	PatLib.Asm	Exec	Edit
Link	PatLib.Link	Exec	Edit
Rmaker	PatLib.R	Exec	Edit

LIBinit.Asm

```
LIBVER_Result      EQU  6

        CLR      LIBVER_Result(A0)
        MOVEQ    #0,D0
        RTS
```

PatLib.R

```
PatLib.Rscr
BLIB
TYPE CODE = GNRL
LIBinit,1
.R
PatLib CODE 1

TYPE CODE = GNRL
PatLib,2
.R
PatLib CODE 2
```

MacPaintAstuces *Suite...*

- Pour entrer du texte continu en modifiant la police, la taille ou le style (voir ci-dessous), faire "Enter" (pas "Return") avant chaque modification.

Changement de taille, de
style (**gras**, souligné)
ou de police avec Enter.

- Quand on maintient la touche Commande enfoncée, tout décor appliqué avec la brosse se superpose en transparent aux décors antérieurs. Si le nouveau décor est plus clair, transiter par une inversion d'image.
- Lorsqu'on a choisi l'option "Editer un motif" on peut éditer n'importe quel décor prélevé par simple clic dans la fenêtre MacPaint.
- Pour quitter MacPaint et éjecter d'un coup la disquette, maintenir la touche Commande enfoncée pendant le choix de l'option Quitter.

Une corbeille dans la

Jean-Luc Bazanegue

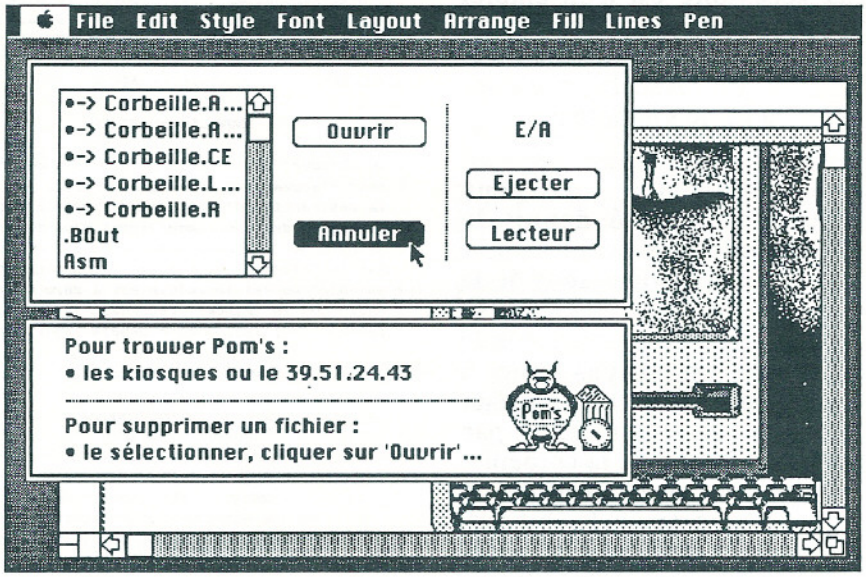
Nous avons tous vu apparaître, sur l'écran du Macintosh, les messages *"Le disque est saturé. Recommencez."*, *"Le disque est saturé. Effacez des documents ou changez de disque."*, ou encore *"Place disponible insuffisante pour enregistrer. Utilisez un autre disque."*. Ce ne sont bien sûr que quelques exemples de ce qui peut se présenter lorsque la disquette de démarrage est saturée, ou proche de la saturation.

La seule solution consiste alors à retourner au "Finder" pour effectuer les manipulations nécessaire, ce qui se traduit, dans la plupart des cas, par la destruction d'un ou plusieurs fichiers devenus inutiles (bout d'essai, ancienne version du fichier traité, etc...). Tout cela fait perdre du temps, et il nous fallait nous mettre à la recherche d'une solution.

Voici donc un petit accessoire de bureau propre à pallier le manque de "corbeille" lorsqu'on utilise un programme. L'accessoire ici proposé est donc une "corbeille de bureau", qui a le même rôle que son homonyme du "Finder", mais, et nous le verrons un peu plus loin, avec des caractéristiques un peu plus "expéditives" (dans un premier temps, nous avions baptisé la chose "Destructeur", mais nous avons préféré remplacer ce nom quelque peu terrorisant par "•-> Corbeille").

Mode d'emploi

L'accessoire en question se trouve dans le fichier "système" de la disquette Pom's Mac 21; vous pouvez donc utiliser directement ce fichier ou, si vous



•-> Corbeille.Asm

```
; Routines utilisées par l'accessoire.

.Trap    _Pack3          $A9EA        ; Appel du "Standard file package".
.Trap    _Delete         $A009        ; Destruction d'un fichier.
.Trap    _GetVolInfo     $A007        ; Lecture des informations sur les volumes.
.Trap    _OffLine        $A035        ; Mise "hors ligne" d'un volume.
.Trap    _MountVol       $A00F        ; Mise "en ligne" d'un volume.
.Trap    _FlashMenuBar   $A94C        ; Inversion (vidéo) d'un menu.
.Trap    _SysBeep        $A9C8        ; Provoque un "Bip" ou une inversion si volume=0.
.Trap    _CopyBits       $A8EC        ; A peu près équivalent au "PUT" du Basic.
.Trap    _SetPort        $A873        ; Indique le port graphique courant.
.Trap    _DisposWindow   $A914        ; Libère la mémoire utilisée par une fenêtre.
.Trap    _NewWindow      $A913        ; Pour créer une nouvelle fenêtre.
.Trap    _DrawString     $A884        ; Affiche une chaîne de caractères.
.Trap    _MoveTo         $A893        ; Déplace le "crayon" à la position indiqué.
.Trap    _LineTo         $A891        ; Trace une ligne depuis la position du crayon.
.Trap    _PenPat         $A89D        ; Détermine le "motif" du "crayon".
.Trap    _TextFont       $A887        ; Police de caractères à utiliser.
.Trap    _BeginUpdate    $A922        ; Début de mise à jour d'une fenêtre.
.Trap    _EndUpdate      $A923        ; Fin de mise à jour d'une fenêtre.

; Décalages par rapport à la base du tampon "TamponIO" de 80 octets, utilisé par les routines
; _Delete, _GetVolInfo, _OffLine et _MountVol pour le stockage des paramètres.

ioCompletion     EQU     $C           ; Pointeur pour entrées/sorties asynchrones.
ioResult         EQU     $10          ; 2 octets pour un éventuel code d'erreur.
ioFileName       EQU     $12          ; Pointeur sur le nom de fichier.
ioVRefNum        EQU     $16          ; 2 octets pour le numéro de référence du volume.
ioFileType       EQU     $1A          ; 1 octet pour le type de fichier.
ioVQElSize       EQU     $40          ; Taille du tampon (64 octets).
ioVNPtr          EQU     $12          ; Pointeur sur le nom de volume.
ioVolIndex       EQU     $1C          ; 2 octets pour le numéro d'index du volume.

; Décalages par rapport à la base du tampon "TamponRep" de 72 octets, utilisé par la routine
; _Pack3 pour le retour des informations.

rGood            EQU     0            ; Drapeau: 0 si "Annuler", 1 si "Ouvrir".
rVolume          EQU     6            ; Numéro de référence du volume.
rVersion         EQU     8            ; Numéro de version du fichier.
rName            EQU     10           ; Nom du fichier (Maximum 63 octets). Le premier
                                      ; octet indique la longueur du nom.

; Equivalences diverses.

sfGetFile        EQU     2            ; Sélecteur de routine (pour _Pack3).
FinderName       EQU     $2E0         ; Adresse de stockage du nom du "Finder".
```

désirez placer "•-> Corbeille" dans un fichier système résidant sur une de vos disquettes, il est bien sûr possible de l'installer avec l'utilitaire "Font/DA Mover".

Cet accessoire est d'un emploi très simple; un mode d'emploi au sens propre du terme serait donc superflu. En revanche, quelques mises en garde s'imposent :

- **"•-> Corbeille" ne peut pas être appelé depuis le "Finder"**; ce serait inutile puisque nous disposons de la corbeille et, de plus, dangereux, étant donné que le "Finder" n'est pas apte à reconnaître la destruction d'un fichier autrement que par l'intermédiaire de la corbeille. Une tentative d'ouverture de l'accessoire depuis le "Finder" se soldera par un "Bip" ou, si le volume sonore est à 0, par une inversion de la barre des menus.
- **"•-> Corbeille" peut détruire n'importe quel fichier**, même s'il est verrouillé, et quel que soit son type. L'accessoire refuse seulement de détruire les fichiers ouverts (le fichier système, l'application en cours ainsi que les éventuels fichiers qui lui sont liés et le fichier en cours de traitement). Si l'on essaie de détruire un fichier ouvert, la barre des menus clignote rapidement pendant une petite seconde pour signaler le problème, et nous revenons à l'application en cours.

Les solutions retenues sont certainement discutables, en particulier la possibilité de détruire à peu près n'importe quoi, mais nous voulions un accessoire efficace qui n'oblige pas l'utilisateur à cliquer sur une série de boutons "OK" ou "Annuler" pour parvenir à faire un peu de place sur une disquette.

```
CurApName       EQU     $910            ; Adresse de stockage du nom de l'application en
                                        ; cours.
dCtlWindow      EQU     $1E             ; Pointeur sur la fenêtre de l'accessoire.
dCtlRefNum      EQU     $18             ; Numéro de référence de l'accessoire.
WindowKind      EQU     $6C             ; Décalage par rapport à la base de
                                        ; l'enregistrement d'une fenêtre.
dBoxProc        EQU     1               ; Numéro de référence des fenêtres type "Dialog".
gray            EQU     $FFFFFFE8       ; Adresse relative (registre A5) du "gris moyen".

; En-tête du driver (voir "Lucy in the Sky with Diamonds", numéro 19 de Pom's).

Base            DC      $400,0,0,0
                DC      Ouverture-Base,Ouverture-Base,Ouverture-Base
                DC      Ouverture-Base,Ouverture-Base
                DC.B    9,'ACC3POM''S'

Ouverture       MOVEM.L A1-A4,-(SP)     ; Sauvegarde les registres d'adresse A1, A2, A3
                                        ; et A4 au sommet de la pile.
                MOVE.L  A1,A4           ; Copie du registre A1 dans le registre A4.

; Le "Finder" autorise la destruction des fichiers par l'intermédiaire de la corbeille, ce qui
; rend l'accessoire inutile lorsque le "Finder" tourne. De plus, le "Finder" ne reconnait pas
; la destruction d'un fichier si elle n'est pas effectuée par la voie normale, ce qui peut
; provoquer des problèmes (appel d'un fichier reconnu par le "Finder", bien que venant d'être
; détruit par l'accessoire). Il est donc prudent de s'assurer que l'application en cours n'est
; pas le "Finder". Si tel est le cas, une tentative d'ouverture de l'accessoire se traduit par
; un "Bip" ou, si le volume est à zéro, par une inversion vidéo de la barre des menus. Notons
; enfin que le "Finder" reconnait la destruction d'un (ou de plusieurs) fichier lorsque l'on
; revient de l'application depuis laquelle l'accessoire a été utilisé.

                LEA     FinderName,A0   ; L'adresse où l'on trouve le nom du "Finder" est
                                        ; placée dans A0.
                LEA     CurApName,A1    ; L'adresse où l'on trouve le nom de
                                        ; l'application en cours est placée dans A1.
                MOVE.B  (A0)+,D1        ; Le premier octet contient le nombre de
                                        ; caractères dans le nom du "Finder".
                CMP.B   (A1)+,D1        ; Il est comparé au nombre de caractères dans le
                                        ; nom de l'application en cours.
                BNE.S   OK              ; Si les deux noms ont des longueurs différentes,
                                        ; l'application en cours n'est pas le "Finder",
                                        ; et on passe à la suite des opérations.
; Si la longueur du nom de l'application en cours est égale à la longueur du nom du "Finder",
; on exécute une boucle qui compare les caractères des deux chaînes. Au stade actuel, l'octet
; de poids faible de D1 contient le nombre de caractères, A0 pointe sur le premier caractère
; du nom du "Finder", A1 pointe sur le premier caractère du nom de l'application en cours.

                EXT     D1              ; Le bit 7 de D1 est copié dans les bits 8 à 15.
                                        ; Cela a pour effet de mettre à 0 l'octet de
                                        ; poids fort du mot de poids faible de D1, qui
                                        ; sera utilisé comme index de boucle.
                SUBQ    #1,D1           ; D1 = nombre de caractères - 1.
BoucleVerif     CMPM.B  (A0)+,(A1)+     ; On compare deux caractères.
                DBNE    D1,BoucleVerif  ; S'ils sont différents, on sort de la boucle.
                                        ; Sinon, on va comparer les deux caractères
                                        ; suivants, sauf si tous les caractères ont été
                                        ; comparés, auquel cas on sort aussi de la
                                        ; boucle.
                ADDQ    #1,D1           ; Si tous les caractères ont été comparés, on
                                        ; trouve -1 dans D1, ce qui signifie que les deux
                                        ; chaînes sont identiques.
                BNE.S   OK              ; Sinon, vers suite des opérations.

                MOVE    #7,-(SP)        ; L'application en cours est le "Finder". On émet
                _SysBeep                ; un "Bip" de 7/60 seconde et on retourne au
                BRA     Sortie          ; "Finder".

; Inversion vidéo de la barre des menus. Elle restera en inverse jusqu'à ce que l'utilisateur
; choisisse un fichier ou "clique" sur "Annuler" (ou "Cancel"). La routine _FlashMenuBar est
; utilisée. On empile soit le numéro du menu à inverser (1 pour le menu "Pomme", par exemple),
; soit 0 pour inverser toute la barre, comme c'est le cas ici.

OK              CLR     -(SP)           ; Deux octets à 0 au sommet de la pile.
                _FlashMenuBar           ; Appel de la routine.

; Création d'une fenêtre du type "boîte à dialogue"

                CLR.L   -(SP)           ; 4 octets pour retour du pointeur.
                CLR.L   -(SP)           ; Stockage de la fenêtre dans le "heap".
                PEA     RectangleF      ; Empile l'adresse des coordonnées de la fenêtre.
                CLR.L   -(SP)           ; Fenêtre sans titre.
                MOVE    #$100,-(SP)     ; Drapeau indiquant que la fenêtre est visible.
                MOVE    #dBoxProc,-(SP) ; Fenêtre du type "boîte à dialogue".
                MOVEQ.L #-1,D0          ; $FFFFFFFF (-1) dans le registre de données D0.
                MOVE.L  D0,-(SP)        ; empile -1 pour fenêtre au premier plan.
                CLR     -(SP)           ; Pas de case "d'échappement".
                CLR.L   -(SP)           ; Pas de procédure spéciale (fenêtre standard).
                _NewWindow              ; Création de la nouvelle fenêtre.
                MOVEA.L (SP)+,A0        ; Dépile pointeur sur la fenêtre, placé dans A0.
                MOVE.L  A0,dCtlWindow(A4)  ; Sauvegarde le pointeur dans le "DCE".
                MOVE    dCtlRefNum(A4),WindowKind(A0)  ; Indique fenêtre "système".
                MOVE.L  A0,-(SP)        ; Empile le pointeur sur la fenêtre.
```

```
            _SetPort                        ; La fenêtre devient le port graphique courant.
            MOVE.L   dCtlWindow(A4),-(SP)   ; Empile le pointeur sur la fenêtre.
            _BeginUpdate                    ; Début de mise à jour de la nouvelle fenêtre.
            CLR      -(SP)                  ; Empile identificateur de la police Chicago (0).
            _TextFont                       ; Chicago devient la police courante.
            MOVE.L   #$D0010,-(SP)          ; Déplace le "crayon" en 16 (coordonnée
            _MoveTo                         ; verticale) et 13 (coordonnée horizontale).
            PEA      'Pour trouver Pom''s : '              ; L'instruction PEA, lorsqu'on
            _DrawString                                    ; lui communique directement une
            MOVE.L   #$1D0010,-(SP)                        ; chaîne de caractères, empile
            _MoveTo                                        ; l'adresse de la chaîne précédée
            PEA      '• les kiosques ou le 39.51.24.43 '   ; d'un octet indiquant sa
            _DrawString                                    ; longueur (indispensable à la
            MOVE.L   #$3D0010,-(SP)                        ; routine "_DrawString" qui se
            _MoveTo                                        ; chargera de l'affichage de la
            PEA      'Pour supprimer un fichier :'         ; chaîne). Au moment de
            _DrawString                                    ; l'assemblage, la chaîne et
            MOVE.L   #$4D0010,-(SP)                        ; l'octet indiquant sa longueur
            _MoveTo                                        ; sont placés à la fin du code.
            PEA      '• le sélectionner, cliquer sur ''Ouvrir''… '
            _DrawString
            PEA      gray(A5)               ; Le registre A5 pointe sur les variable globales
            _PenPat                         ; parmi lesquelles on trouve la définition des
            MOVE.L   #$290010,-(SP)         ; "grisés". Le "gris moyen" se trouve à l'adresse
            _MoveTo                         ; pointée par A5, moins 24.
            MOVE.L   #$290108,-(SP)         ; Trace une ligne pointillée entre la position
            _LineTo                         ; courante du crayon et les coordonnées indiquées

; Affichage de la "bestiole"

            LEA      BitMap,A6              ; Adresse de la structure "BitMap" dans A6.
            LEA      Def,A3                 ; Adresse de la définition du dessin dans A3.
            MOVE.L   A3,(A6)                ; Adresse de la définition dans la structure.
            MOVE.L   A6,-(SP)               ; Empile l'adresse de la structure "BitMap".
            MOVEA.L  dCtlWindow(A4),A6      ; Le pointeur sur la fenêtre est placé dans A6.
            ADDQ.L   #2,A6                  ; Ajoute 2 au pointeur pour "BitMap" fenêtre.
            MOVE.L   A6,-(SP)               ; Adresse du "BitMap" de la fenêtre dans A6.
            PEA      RectSource             ; Empile l'adresse du rectangle source.
            PEA      RectDest               ; Empile l'adresse du rectangle destination.
            CLR      -(SP)                  ; Transfert classique (PENMODE 0).
            CLR.L    -(SP)                  ; Pas de masque.
            _CopyBits                       ; Affichage.

            MOVE.L   dCtlWindow(A4),-(SP)   ; Empile le pointeur sur la fenêtre.
            _EndUpdate                      ; Fin de mise à jour de la fenêtre.
; Appel et affichage de la fenêtre standard de sélection des fichiers. Pour appeler la
; routine, employée ici de la manière la plus simple, il faut empiler, dans l'ordre :
; - la position horizontale de l'angle supérieur gauche de la fenêtre (2 octets);
; - la position verticale de l'angle supérieur gauche de la fenêtre (2 octets);
;                   (positions données en coordonnées globales)
; - 4 octets à 0 (jamais utilisés);
; - un pointeur sur une procédure de filtrage des fichiers (non utilisé);
; - nombre de types indiquant les fichiers à afficher, ou -1 pour afficher tous les fichiers
;                   (2 octets);
; - pointeur sur la liste des types de fichier, ou 0 si zone précédente = -1;
; - pointeur en cas d'utilisation d'une fenêtre personnalisée, 0 sinon;
; - adresse d'un tampon de 72 octets utilisé par la routine pour le retour des informations
;                   (nom du fichier, numéro de référence du volume...);
; - sélecteur de routine (2 octets), 2 pour notre accessoire.
            MOVE.L   #$00280014,-(SP)       ; Coordonnées.
            CLR.L    -(SP)                  ; 0.
            CLR.L    -(SP)                  ; Pas de filtrage.
            MOVE     #-1,-(SP)              ; Tous les fichiers.
            CLR.L    -(SP)                  ; Pas de liste de types.
            CLR.L    -(SP)                  ; Fenêtre standard.
            PEA      TamponRep              ; Adresse du tampon "TamponRep".
            MOVE     #sfGetFile,-(SP)       ; Selecteur.
            _Pack3                          ; Appel de la routine.

            LEA      TamponRep,A1           ; Adresse de base du tampon dans A1.
            TST      rGood(A1)              ; Si nul, l'utilisateur a "cliqué" sur
            BEQ.S    Annuler                ; "Annuler": retour à l'application.

            LEA      TamponIO,A0                          ; Les routines
            CLR.L    ioCompletion(A0)                     ; utilisée pour les
            LEA      rName(A1),A2                         ; entrées/sorties
            MOVE.L   A2,ioFileName(A0)                    ; ont été décrites
            MOVE     rVolume(A1),ioVRefNum(A0)            ; dans les numéros
            MOVE.B   rVersion(A1),ioFileType(A0)          ; 17 ("Catalogue sur
            _Delete                                       ; Imprimante", "BSAVE
            TST      D0                                   ; et BLOAD avec le
            BEQ.S    Suite                                ; Basic Microsoft") et
                                                          ; 18 ("Modifications
            MOVEQ    #19,D3                               ; des informations
BoucleFlash CLR      -(SP)                                ; pour le "Finder"") de
            _FlashMenuBar                                 ; Pom's.
            DBRA     D3,BoucleFlash                       ; TST D0 après
            BRA.S    Annuler                              ; _Delete permet de
                                                          ; savoir si le fichier
Suite       CLR.L    ioCompletion(A0)                     ; est en cours
            CLR      ioVolIndex(A0)                       ; d'utilisation.
```

Nous ne nous étendrons pas sur son fonctionnement, le source (aussi présent sur la disquette 21) étant généreusement commenté.

MacAstuces

Tous les fichiers créés sur une disquette, même s'ils sont mis à la poubelle et si celle-ci est vidée, laissent une trace sur votre disquette et occupent encore de la place. Pour y remédier :

- Initialisez une disquette,
- Recopiez tous les fichiers sur cette nouvelle disquette. Vous aurez toujours vos fichiers, mais certainement avec de précieux Ko supplémentaires.

Après de nombreuses opérations de destruction de fichiers, on a donc intérêt à "rafraîchir" la disquette en recopiant tous ces fichiers sur une nouvelle disquette, quitte à recopier ensuite celle-ci entièrement sur l'original afin de garder la belle étiquette qu'on y avait collée...

Un bug dans la LaserWriter !

"PostScript error" suivi du message peu réconfortant "Document OK, but cannot be printed" : voilà ce qui arrive lorsqu'on tente d'imprimer sur la Laser Writer un document dont le nom comporte une parenthèse fermante.
Merci à Bruno Rives qui nous a aidés dans l'analyse, pas évidente, de ce bug !

```
            LEA       rName(A1),A2
            MOVE.L    A2,ioVNPtr(A0)          ; Si l'utilisateur de
            _GetVolInfo                       ; l'accessoire tente de
            _OffLine                          ; détruire un fichier
            MOVE      #1,D1                   ; ouvert, on inverse
Erreur      MOVE      D1,ioVRefNum(A0)        ; rapidement, et 20
            _MountVol                         ; fois de suite, la
            MOVE      #2,D1                   ; barre des menus.
            TST       ioResult(A0)            ; On retourne ensuite à
            BNE.S     Erreur                  ; l'application.

; Avant de revenir à l'application, on inverse une dernière fois la barre
; des menus, afin qu'elle ne reste pas en caractères blancs sur fond noir.

Annuler     MOVE.L    dCtlWindow(A4),-(SP)
            CLR.L     dCtlWindow(A4)
            _DisposWindow
            CLR       -(SP)
            _FlashMenuBar
Sortie      MOVEM.L   (SP)+,A1-A4
            CLR       D0
            RTS

; Tampons et contantes.

TamponIO    DCB.B     ioVQElSize,0    ; 64 octets pour entrées/sorties.
TamponRep   DCB       36,0            ; 72 octets pour _Pack3.
RectangleF  DC        198,20,281,368  ; Coordonnées (globales) fenêtre.
; Structure "BitMap".
BitMap      DC.L      0               ; Stockage adresse de la définition.
OLBitMap    DC        8               ; Nombre d'octets par ligne.
RectSource  DC        0,0,54,64       ; Coordonnées du rectangle source.

RectDest    DC        18,280,72,344   ; Coordonnées rectangle destination.

Def   DC   $0008,$0002,$0000,$0000,$0004,$0004,$0000,$0000,$0006,$000C,$0000,$0000,$0006,$000C,$0000,$0000,$0006
      DC   $000C,$0000,$0000,$0005,$3F94,$0000,$0000,$0002,$C068,$0000,$0000,$0001,$0010,$0000,$0000,$0002,$0008
      DC   $0000,$0000,$0002,$7FC8,$0000,$0000,$0004,$BBA4,$0000,$0000,$0004,$1F04,$0000,$0000,$0004,$2084,$0000
      DC   $0000,$001E,$404F,$0000,$0000,$006A,$404A,$C000,$FC00,$01DF,$209F,$7001,$4A00,$02B3,$DF79,$A802,$3100
      DC   $0561,$FFF0,$D404,$0280,$0AD4,$FFE5,$6A09,$2440,$1580,$3F80,$3512,$4120,$2B00,$0000,$1AA4,$9210,$3600
      DC   $0000,$0D90,$2488,$6C00,$5540,$06CA,$4924,$5415,$0015,$0544,$9002,$A940,$0000,$52A3,$FFFF,$D81C,$0000
      DC   $0362,$0001,$A832,$2A80,$02A2,$4445,$D032,$0008,$0162,$8889,$B132,$0018,$11A2,$8889,$5432,$67C3,$8542
      DC   $8889,$703C,$D6A6,$01C2,$8889,$3030,$D6A3,$0182,$8889,$3130,$D6A1,$8182,$8889,$1830,$D6A0,$8302,$8889
      DC   $0830,$66A7,$0202,$9F89,$0A00,$0000,$0A02,$E069,$04A8,$0002,$A403,$8519,$0402,$AAA8,$0402,$2045,$0200
      DC   $0000,$0C02,$8015,$0328,$0002,$9804,$0003,$0182,$2A88,$3005,$000B,$00C2,$4048,$6008,$0801,$0064,$8024
      DC   $C00A,$1C05,$003C,$8027,$8008,$0E01,$0018,$E0E3,$000A,$0705,$0009,$FFF2,$0008,$0381,$0009,$FFF2,$0008
      DC   $0101,$0004,$FFE4,$0005,$000B,$0004,$9F24,$0004,$0003,$00E2,$60C8,$E002,$8015,$0317,$515D,$1802,$2045
      DC   $1C08,$1102,$0703,$8A19,$6000,$2080,$00C2,$6061,$3FFF,$C07F,$FF83,$FFFF
```

•-> Corbeille.R

```
•-> Corbeille.Rsrc

TYPE DRVR = PROC
•-> Corbeille ,30
•-> Corbeille
```

•-> Corbeille.Job

```
Asm     •-> Corbeille.Asm    Exec   Edit
Link    •-> Corbeille.Link   Exec   Edit
Rmaker  •-> Corbeille.R      Exec   Edit
```

•-> Corbeille.Link

```
[

•-> Corbeille.Rel

$
```

Disquette Macintosh numéro 21

Outre les fichiers et programmes listés dans ce numéro, vous trouverez sur cette disquette les polices ***TimesBold*** *(10, 12, 14, 18 et 24 points),* ***TimesBoldItalic*** *(10, 12, 14, 18 et 24 points),* ***TimesItalic*** *(10, 12, 14, 18 et 24 points),* ***HelveticaBold*** *(10, 12, 14, 18 et 24 points),* ***HelveticaBoldOblique*** *(10, 12, 14, 18 et 24 points),* ***HelveticaOblique*** *(10, 12, 14 et 18 points),* ***CourierBold*** *(10 et 12 points),* ***CourierBoldOblique*** *(10 et 12 points) et* ***CourierOblique*** *(10 et 12 points).*

Ces polices remplacent avantageusement, lorsqu'elle sont employées en gras ou en italique (ou les deux), celles de l'imprimante LaserWriter. Les caractères affichés à l'écran sont beaucoup plus lisibles et l'impression (sur ImageWriter) plus fidèle.

Exemple en TimesBold 12 :

Le programme source étant largement commenté, nous nous bornerons à indiquer ici les caractéristiques principales du déroulement de la routine.

Exemple en Times gras 12 :

Le programme source étant largement commenté, nous nous bornerons à indiquer ici les caractéristiques principales du déroulement de la routine.

Fenêtres de saisie

Gérard Michel

Cet article n'enrichira guère votre Basic, qui dispose déjà d'instructions (telles que WINDOW, EDIT FIELD ou EDIT$) pour vous permettre la saisie de variables via fenêtres et gestion de texte.

En revanche, peut-être vous apportera-t-il quelques éclaircissements sur la façon dont s'opèrent la manipulation des fenêtres et l'édition de textes au niveau du système, et donc sur le type de routines utilisées par le Basic.

Les opérations à réaliser dans le cadre du programme en assembleur listé plus loin sont les suivantes :

- Ouverture d'une fenêtre dont on donne les limites, via un tableau du Basic.
- Ouverture d'un "texte éditable", dont on indique de même les limites "visibles" et les dimensions.
- Saisie de texte.
- Utilisation de la souris pour déplacer le curseur, placer un point d'insertion de texte, ou sélectionner un "morceau" de texte en vue de l'effacer.
- Retour au Basic lors de la frappe de la touche RETURN ou de la touche "commande" (et ce quelle que soit la position du curseur à cet instant).
- Récupération du texte dans une variable chaîne de caractères, dans la limite d'un nombre maximum de caractères indiqué au départ par la longueur de la chaîne.

A partir du Basic, cette routine s'appelle de manière classique, avec passage en paramètre du pointeur sur le descripteur de la chaîne utilisée pour le transfert (voir programme-exemple).

Principes de fonctionnement

Le programme source étant largement commenté, nous nous bornerons à indiquer ici les caractéristiques principales du déroulement de la routine.

Variables de travail

• Lorsque se produit un "événement" (enfoncement d'une touche, déplacement de la souris, clic...), il se caractérise par un certain nombre d'informations : type de l'événement, contenu, résultat, type de touche ou de "clic"... Ces informations sont retournées dans une zone de stockage dont l'adresse doit être indiquée à la routine chargée de saisir les événements *(GetNextEvent)*. La zone baptisée *Evenement* est utilisée pour cela.

• Un texte éditable comporte une zone de visibilité *(ViewRect* dans notre exemple*)* et une zone de stockage qui lui est au moins égale *(DestRect)*.

• *FenetreStockage* est réservée pour le stockage de l'ensemble des informations nécessaires à la gestion d'une fenêtre par le système.

Principales routines utilisées

• *NewWindow*, comme son nom l'indique, permet d'ouvrir une nouvelle fenêtre dont on indique les paramètres, en particulier les dimensions, le type et le titre éventuel... Cette fonction retourne un pointeur sur la fenêtre, qui permettra ensuite d'appeler les fonctions de gestion des fenêtres, par exemple *SetPort* qui spécifie le "port" d'entrée/sortie à retenir par les autres routines.

• *TENew* ouvre un enregistrement "texte éditable", dont on indique au préalable les zones de visibilité et de stockage. Cette fonction retourne un "Handle" (c'est-à-dire un pointeur sur l'adresse d'un pointeur) sur l'enregistrement texte ainsi constitué, "Handle" qui permettra l'appel des autres routines d'édition de texte. *TEActivate* active l'édition de texte, tandis que *TEIdle* fait apparaître le curseur et détermine ainsi le point d'insertion du texte.

• *TEGetText*, moyennant l'indication du "Handle" sur l'enregistrement texte, retourne un "Handle" sur les caractères qui le constituent, autorisant finalement l'accès au contenu même du texte, en termes de codes ASCII. A noter par ailleurs que les deux octets qui se trouvent à $3C et $3D du début de l'enregistrement texte contiennent la longueur du texte, et donc le nombre de caractères.

• *TEKey* assure pour sa part l'édition d'un caractère, soit affichage et stockage dans le texte (il faut lui passer en paramètres, via la pile, le code ASCII du caractère en question et le "Handle" sur le texte concerné). *TEClick* prend en charge les "clics" dans le texte (insertion du curseur ou sélection d'une zone).

• *GlobalToLocal* convertit les coordonnées de la souris retournées par *GetNextEvent*, calculées par rapport à l'écran dans son ensemble, dans le système d'axes déterminé par la fenêtre active, afin d'assurer l'insertion à l'endroit voulu.

• Les autres routines, relatives à la manipulation des fenêtres, parlent d'elles-mêmes...

Programme de démonstration

```
DEFINT A-Z
VV$=SPACE$(256)
DIM C(323),d$(4)
DATA 0,0,0,0,0,0,0,0,0,0
' Rectangles de la fenêtre et du texte
DATA 50,40,70,200,3,2,18,150,3,2,18,150
' Début du code de la routine
DATA &H4E56,0,&H42A7,&HA924,&H261F
DATA &H42A7,&H487A,&HFEBA,&H487A
DATA &HFE9E,&H42A7,&H1F3C,-1,&H3F3C,1
DATA &H2F3C,-1,-1,&H1F3C,0,&H2F3C,0,0
DATA &HA913,&H2657,&HA873,&H42A7,&H487A
DATA &HFE88,&H487A,&HFE7C,&HA9D2
DATA &H2457,&HA9D8,&H2F0A,&HA9DA
DATA &H4267,&H3F3C,&H0FFF,&H487A,&HFE4C
DATA &HA970,&H301F,&H67EC,&H6046,&H2F0A
DATA &HA9CB,&H285F,&H2254,&H2852,&H322C
DATA &H003C,&H206E,8,&H1828,1,&H2028,1
DATA &H0280,&H00FF,-1,&H2840,&H7000
DATA &HB041,&H6700,10,&H5240,&H18D9
DATA &HB800,&H66F2,&H2F0A,&HA9CD
DATA &H2F0B,&HA92D,&H2F03,&HA91F,&H2F03
DATA &HA873,&H4E5E,&H4E75,&H383A
DATA &HFE08,&H303A,&HFDF6,&H0C40,1
DATA &H6700,&H0032,&H0C40,3,&H6700,12
DATA &H0C40,5,&H6700,4,&H6682,&H0804,8
DATA &H6692,&H3F3A,&HFDD6,&H301F,&H0C00
DATA 13,&H6786,&H3F3A,&HFDCA,&H2F0A
DATA &HA9DC,&H4EFA,&HFF66,&H4267
DATA &H2F3A,&HFDC2,&H487A,&HFDC4
DATA &HA92C,&H301F,&H0C40,3,&H6600
DATA &HFF50,&H42A7,&HA924,&H201F,&HB08B
DATA &H6700,12,&H2F3A,&HFDA8,&HA91F
DATA &H4EFA,&HFF3A,&H487A,&HFD98
DATA &HA871,&H2F3A,&HFD92,&H103C,0
DATA &H1F00,&H2F0A,&HA9D4,&H4EFA,&HFF22
FOR i=0 TO 21:READ c(i):NEXT
' zone de stockage de la fenêtre
FOR i=22 TO 177:c(i)=0:NEXT
FOR i=178 TO 323:READ c(i):NEXT:DE=178
10 CLS:V$=LEFT$(vv$,15):A!=VARPTR(c(DE))
A! VARPTR(V$)
d$(1)=v$ ' v$ comporte 15 caractères, dont des
' blancs à la fin si nécessaire
' déplacement de la fenêtre
c(10)=80:c(11)=40:c(12)=100:c(13)=200
v$=LEFT$(vv$,15):A!=VARPTR(c(DE))
A! VARPTR(V$)
d$(2)=v$
' déplacement et changement de dimension
c(10)=100:c(11)=20:c(12)=250:c(13)=450:c(14)=3
c(15)=4:c(16)=130:c(17)=420:c(18)=3:c(19)=4
c(20)=130:c(21)=420
V$=LEFT$(vv$,255):A!=VARPTR(c(DE))
A! VARPTR(V$)
d$(3)=v$
c(10)=80:c(11)=240:c(12)=100:c(13)=400:c(15)=2
c(16)=18:c(17)=150:c(19)=2:c(20)=18:c(21)=150
v$=LEFT$(vv$,15):A!=VARPTR(c(DE))
A! VARPTR(V$)
d$(4)=v$
CLS:FOR i=1 TO 4:PRINT d$(i)
PRINT LEN(d$(i)):NEXT
INPUT z$
IF z$="" THEN c(10)=50:c(11)=40:c(12)=70:c(13)
    =200:GOTO 10
```

Source de la routine en langage machine

```
.Trap    _NewWindow      $A913
.Trap    _SetPort        $A873
.Trap    _TEIdle         $A9DA
.Trap    _GetNextEvent   $A970
.Trap    _TENew          $A9D2
.Trap    _TEActivate     $A9D8
.Trap    _TEGetText      $A9CB
.Trap    _TEDispose      $A9CD
.Trap    _TEKey          $A9DC
.Trap    _FindWindow     $A92C
.Trap    _FrontWindow    $A924
.Trap    _SelectWindow   $A91F
.Trap    _GlobalToLocal  $A871
.Trap    _TEClick        $A9D4
.Trap    _CloseWindow    $A92D
```

```
                        Evenement       ; Caractéristiques de l'événement
0000                    Quoi:           DC      0               ; Type
0000 0000               Message:        DC.L    0               ; Information complémentaire
0000 0000               Quand:          DC.L    0               ; Moment
0000 0000               Point:          DC.L    0               ; Coordonnées Souris
0000                    Modifie:        DC      0               ; Etat touche et bouton
0000 0000               WFenetre:       DC.L    0               ; Résultat de recherche de fenêtre

0032 0028 0046 00C8     Rectangle       DC      50,40,70,200    ; Rectangle de la fenêtre
0003 0002 0012 0096     ViewRect        DC      3,2,18,150      ; Rectangle de visibilité du texte
0003 0002 0012 0096     DestRect        DC      3,2,18,150      ; Limites du texte

xxxx xxxx xxxx          FenetreStockage DCB     $9C,0
                        ModifieReg      EQU     D4              ; Bits de modif. par GetNextEvent
                        TextHReg        EQU     A2              ; "Handle" sur le texte
                        FenetrePReg     EQU     A3              ; "Handle" sur la fenêtre
                        CMDKEY          EQU     8               ; Signale usage touche "commande"

4E56 0000                               LINK    A6,#0
42A7                                    CLR.L   -(SP)
A924                                    _FrontWindow
261F                                    MOVE.L  (SP)+,D3        ; Sauvegarde Handle fenêtre initiale
                        MAJFenetre
42A7                                    CLR.L   -(SP)           ; Ouverture d'une nouvelle fenêtre
487A FEBA                               PEA     FenetreStockage
487A FE9E                               PEA     Rectangle
42A7                                    CLR.L-(SP)              ; Sans titre
1F3C FFFF                               MOVE.B  #-1,-(SP)       ; Visible de suite
3F3C 0001                               MOVE    #1,-(SP)        ; De type "boîte à dialogue"
2F3C FFFF FFFF                          MOVE.L  #-1,-(SP)       ; Devant toutes les autres
1F3C 0000                               MOVE.B  #0,-(SP)        ; Sans "carré de fuite"
2F3C 0000 0000                          MOVE.L  #0,-(SP)
A913                                    _NewWindow
2657                                    MOVE.L  (SP),FenetrePReg
A873                                    _SetPort                ; Nouvelle fenêtre = port entrée/so.
                        MAJTextEdit
42A7                                    CLR.L   -(SP)           ; Ouverture d'un texte éditable
487A FE88                               PEA     DestRect
487A FE7C                               PEA     ViewRect
A9D2                                    _TENew
2457                                    MOVE.L  (SP),TextHReg   ; Dont on récupère le Handle
A9D8                                    _TEActivate             ; Et que l'on active
                        Boucle
2F0A                                    MOVE.L  TextHReg,-(SP)
A9DA                                    _TEIdle                 ; Curseur clignotant
4267                                    CLR     -(SP)
3F3C 0FFF                               MOVE    #$0FFF,-(SP)    ; Autorise 12 types d'événements
487A FE4C                               PEA     Evenement
A970                                    _GetNextEvent           ; Un événement ?
301F                                    MOVE    (SP)+,D0
67 EC                                   BEQ     Boucle          ; Non... Attendre...
60 46                                   BRA.S   Traitev         ; Oui... Le traiter
                        Sortie
2F0A                                    MOVE.L  TextHReg,-(SP)  ; Il faut récupérer le texte
A9CB                                    _TEGetText              ; Avant retour au Basic
285F                                    MOVEA.L (SP)+,A4        ; Pointeur/handle sur le texte
2254                                    MOVEA.L (A4),A1         ; A1=adresse des caractères
2852                                    MOVEA.L (A2),A4         ; A4=adr. de l'enregistrement Texte
322C 003C                               MOVE    $3C(A4),D1      ; Longueur du texte
206E 0008                               MOVEA.L 8(A6),A0        ; Récupère le descripteur de chaîne
1828 0001                               MOVE.B  1(A0),D4        ; Longueur de la chaîne (≤ 255)
2028 0001                               MOVE.L  1(A0),D0        ; Adresse de la chaîne
0280 00FF FFFF                          ANDI.L  #$00FFFFFF,D0   ; Convertie en 1 mot long
2840                                    MOVEA.L D0,A4
70 00                                   MOVEQ   #0,D0           ; D0=nombre de caractères copiés
```

```
                    Compare
B041                        CMP     D1,D0           ; Autant de caractères que le
67 00 00 0A                 BEQ     Fini            ; texte en contient ?
5240                        ADDQ    #1,D0
18D9                        MOVE.B  (A1)+,(A4)+     ; Transfert "dans" la chaîne
B800                        CMP.B   D0,D4           ; Autant de caractères que la
66 F2                       BNE     Compare         ; chaîne peut en contenir ?
                    Fini
2F0A                        MOVE.L  TextHReg,-(SP)
A9CD                        _TEDispose              ; Rend place du texte à la machine
2F0B                        MOVE.L  A3,-(SP)
A92D                        _CloseWindow            ; Ferme notre fenêtre
2F03                        MOVE.L  D3,-(SP)
A91F                        _SelectWindow           ; Remet en activité fenêtre initiale
2F03                        MOVE.L  D3,-(SP)
A873                        _SetPort
4E5E                        UNLK    A6              ; Retour au Basic
4E75                        RTS
                    Traitev
383A FE08                   MOVE    Modifie,ModifieReg
303A FDF6                   MOVE    Quoi,D0         ; Quel type d'événement
0C40 0001                   CMPI    #1,D0           ; Clic
67 00 00 32                 BEQ     Souris
0C40 0003                   CMPI    #3,D0           ; Touche
67 00 00 0C                 BEQ     Touche
0C40 0005                   CMPI    #5,D0           ; Auto-répétition d'une touche
67 00 00 04                 BEQ     Touche
66 82                       BNE     Boucle          ; On ne prend pas les autres
                    Touche
0804 0008                   BTST    #CmdKey,ModifieReg      ; Touche commande ?
66 92                       BNE     Sortie          ; Si oui => sortir
3F3A FDD6                   MOVE    Message+2,-(SP)
301F                        MOVE    (SP)+,D0
0C00 000D                   CMPI.B  #13,D0          ; Touche RETURN ?
67 86                       BEQ     Sortie
3F3A FDCA                   MOVE    Message+2,-(SP)  ; Sinon, faire éditer le caractère
2F0A                        MOVE.L  TextHReg,-(SP)
A9DC                        _TEKey
4EFA FF66                   BRA     Boucle
                    Souris
4267                        CLR     -(SP)
2F3A FDC2                   MOVE.L  Point,-(SP)
487A FDC4                   PEA     WFenetre
A92C                        _FindWindow             ; Où s'est produit le clic ?
301F                        MOVE    (SP)+,D0
0C40 0003                   CMPI    #3,D0           ; Dans la zone "contenu" ?
66 00 FF50                  BNE     Boucle
42A7                        CLR.L   -(SP)
A924                        _FrontWindow            ; Si oui, notre fenêtre est-elle
201F                        MOVE.L  (SP)+,D0
B08B                        CMP.L   FenetrePReg,D0 ; toujours active et "au-dessus"
67 00 00 0C                 BEQ     Dsuite
2F3A FDA8                   MOVE.L  WFenetre,-(SP) ; Si non, la ré-activer et attendre
A91F                        _SelectWindow           ; un autre évènement
4EFA FF3A                   BRA     Boucle
                    Dsuite
487A FD98                   PEA Point
A871                        _GlobalToLocal          ; Convertit dans le système de
2F3A FD92                   MOVE.L  Point,-(SP)     ; coordonnées de la fenêtre
103C 0000                   MOVE.B  #0,D0
1F00                        MOVE.B  D0,-(SP)        ; Pas de clic "majuscule"
2F0A                        MOVE.L  TextHReg,-(SP)
A9D4                        _TEClick                ; Fait "éditer" le clic
4EFA FF22                   BRA Boucle
```

La 36ème piste ...ou 528 secteurs disponibles

B. Toméno

Une disquette "normale", formatée sous DOS 3.3, comporte, comme chacun sait, 35 pistes de 16 secteurs, soit 560 secteurs dont seulement 496 sont disponibles pour l'enregistrement des programmes puisque le DOS se réserve 48 secteurs (pistes 00,01,02) et qu'il en alloue encore 16 pour le stockage du catalogue et de la VTOC (piste 17) : voir schéma ci joint.

Or le DOS ne se sert réellement que de 37 des 48 secteurs qu'il se réserve (il utilise la totalité des pistes 00 et 01, mais seulement 5 secteurs de la 02, à savoir les secteurs 00 à 04) . De même CATALOG et VTOC laissent inoccupés 5 des 16 secteurs de la piste 17, à savoir les secteurs 01 à 05. Recupérer les 11 secteurs libres de la 02 et les 5 de la 17, c'est une piste gagnée !

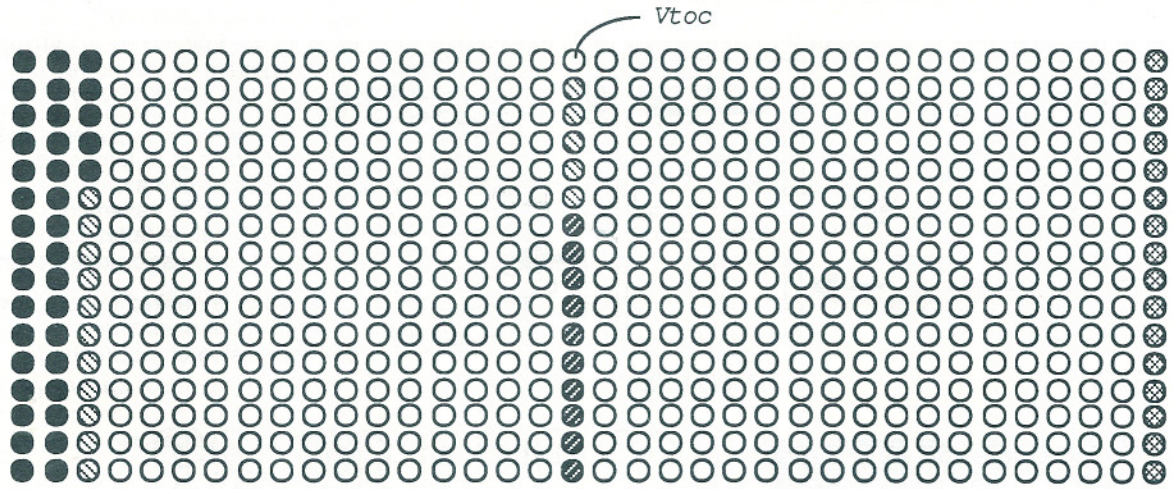
Le formatage en 35 pistes n'est qu'une habitude; en fait il y a sur une disquette de quoi tracer une 36ème piste que 95% des lecteurs sont capables de lire. On peut ainsi encore gagner 16 secteurs et on arrive à 496 + 16 + 16 = 528 secteurs-utilisateurs.

La pratique

Tout d'abord il faut se munir d'un éditeur de secteurs genre DISK-FIX mais tout autre peut convenir (COPY II+ 4.3, NIBBLE AWAYS...) y compris ceux publiés dans ***Pom's*** (voir *Mobby disk* dans n°15 ou *Accès direct* dans le n°16).

La manœuvre se fait en 5 temps (les 4 premiers modifient la VTOC, secteur 00 de la piste 17).

1 Initialiser une disquette avec votre Hello préféré... mais après être passé en moniteur (Call -151) et avoir remplacé la valeur de $BEFE (normalement $23) par $24! Ceci va entrainer un formattage sur 36 pistes lors de l'initialisation. Charger votre éditeur de secteurs et lire le secteur 00 de la piste $11 (17). Sur la 7ème ligne (si votre éditeur affiche des lignes de 8 octets, ce qui est le cas habituel sauf pour NIBBLE AWAYS) l'octet $35 (le 53ème du secteur, le 5ème de la ligne 30) doit être à $24, et non pas à $23 comme sur une disquette normale. (Une autre possibilité pour faire cette première manipulation est de formatter directement une disquette sur $24 pistes avec un utilitaire Ad Hoc puis de remplacer $23 par $24 avec votre éditeur de secteurs sur le secteur



Plan d'occupation des secteurs d'une disquette DOS 3.3

00 de la piste $11).

2 Informer le DOS que cette 36ème piste est inoccupée : toujours sur le secteur 00 de la piste $11 remplacer en ligne $C0 les 5ème et 6ème octets (00 00) par FF FF.

3 Dire au DOS que les secteurs 01 à 05 de la piste $11 sont libres. Toujours sur le même secteur, lire en ligne $78 les 5ème et 6ème octets, normalement à 00 00, et les remplacer par 00 3E.

4 Déclarer que les 11 secteurs ($5 à $0F) de la piste 02 sont disponibles. En ligne $40 du même secteur remplacez les octets 1 et 2 (00 00) par FF E0.
Réécrivez alors le secteur 00 sur la disquette

5 Le 5ème temps se fait en lisant le secteur $06 de la piste $11. Lire ce secteur et sur la 1ère ligne ($00), remplacer par 00 00 les octets 2 et 3 (normalement 11 05, ce qui indique au DOS de poursuivre le CATALOG sur la piste $11, secteur $05, au cas, bien improbable, où le catalogue serait si fourni que les secteurs $06 à $0F ne lui suffiraient pas) : si le DOS veut poursuivre l'écriture du catalogue sur les secteurs 01 à 05 de la piste 17, il est ainsi informé que ceux ci ne sont pas disponibles.

Réécrivez maintenant ce secteur 06 sur la disquette. Voilà, vous avez gagné 36 secteurs soit 2 pistes.

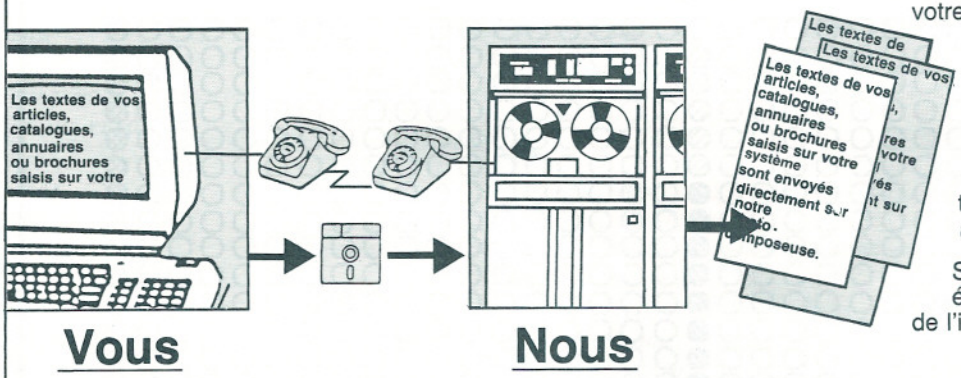
Si vous faites la même chose sur une disquette sans DOS (le cas des disquettes réservées au stockage des fichiers d'un tableur ou d'un traitement de texte par exemple, mais vous pouvez vous les fabriquer vous même avec DISK-FIX ou d'autres programmes permettant de supprimer le DOS) vous obtenez des disquettes de 564 secteurs-utilisateur . 564 et non pas 576 comme le voudrait 16*36, car il faut quand même garder un peu de place pour Catalogue (10 secteurs), VTOC (1 secteur) et message de BOOT éventuellement(secteur 00 de la piste 00).

Evidemment pour copier une disquette ainsi batie il faut prendre un utilitaire capable de reconnaitre la piste 36 : le plus rapide semble être encore " MUNCHER " (moins de 30 secondes !) tandis que Locksmith ou Crazy Copy le feront en quelques minutes (répondre $23 à la question "End track").

Encore une remarque : cette manœuvre est aussi possible avec les DOS non standard (Z-DOS et compagnie) d'autant que certains d'entre eux sont encore plus courts que le DOS normal.

Vos textes en direct de votre ordinateur à nos photocomposeuses

Gain de temps et économie



Les textes de vos articles, catalogues, annuaires ou brochures saisis sur votre micro-ordinateur sont envoyés directement sur notre photocomposeuse

Nous vous évitons ainsi, le coût et le temps de la saisie supplémentaire que nécessite le traitement traditionnel de la photocomposition avant l'impression des documents.

Si vous le désirez nous pouvons également nous charger de l'impression et du brochage.

Vous **Nous**

TELECOMPO (1) 328.18.63

PHOTOCOMPOSITION - BUREAUTIQUE - TRANSMISSION DE DONNÉES - GESTION DE FICHIERS - MATÉRIELS DE TRAITEMENT DE TEXTES

13 et 15, avenue du Petit-Parc - 94300 VINCENNES

Une référence : la revue **pom's**

Un compatible in compatible ?

Yvan Koenig

Ce titre un peu provocateur reflète ma première réaction après examen du kit de mise à niveau //c pour l'Apple //e. On ne trouve en effet pas moins de 1094 octets différents dans les ROM des adresses $D000 à $FFFF qui regroupent l'interpréteur Applesoft et le moniteur.

De plus, les ROM internes en $C100 sont totalement différentes.

Finalement, les différences de codage n'induisent pas de très gros problèmes de compatibilité. La plus grosse déception est due à la fonction LIST du moniteur qui, bien que légèrement recodée, ne reconnaît pas les nouvelles instructions du processeur 65C02.

On retrouve sur //e+ les fonctions de lecture/écriture sur cassette qui avaient été supprimées sur le //c.

En revanche, les routines LORES80 du //c qui permettent de manipuler la basse résolution sur 80 colonnes n'existent pas sur le nouveau //e+. Il auraît peut-être été préférable de les inclure, plutôt qu'un Mini-Assembleur extrait du moniteur du tout premier Apple II qui, lui non plus, ne reconnaît pas les nouveaux opcodes du 65C02.

Certains petits détails peuvent finir par agacer : certaines routines identiques (minMAJ, spLINPRT, EOL?, sbCH) ne sont pas implantées à la même adresse. L'adresse $1F de la page zéro n'est finalement exploitée que par l'ancien Apple //e.

L'octet en $E006, qui valait toujours $00 sous Applesoft, contient un $C4 sur le //e+, ce qui provoque quelques plantages pour les programmes qui utilisaient cet octet comme test de référence (exemple David-DOS) : il vaut mieux désormais référencer l'adresse $E050.

Sur le plan hardware, quelques problèmes d'images graphiques "déchirées" peuvent survenir avec la ROM vidéo du kit de mise à niveau fournie par Apple; si cela vous arrive faites-en une copie sur une EPROM rapide. De même, Apple USA reconnaît que certains chips MMU (références 8314, 8424, 8425, 8426) sont "buggés" et créent des problèmes avec les cartes 80 colonnes étendues.

Les tableaux suivants reprennent l'ensemble des différences relevées entre II+, //e, //e+ (ainsi a-t-on baptisé le //e après mise à niveau) et //c, pour respectivement les bancs d'adresses $D000-$F7FF, $F800-FBFF, $FC00-$FFFF. Afin d'alléger ces tableaux, les conventions suivantes sont appliquées:

- "." indique une instruction identique à celle du II+,
- "+" indique une instruction identique à celle du //e,
- "*" indique une instruction identique à celle du //e+.

***NDLR** : Dans un prochain numéro, nous publierons la routine DESASM/RAM (assembleur Big Mac) d'Yvan Koenig qui permet de désassembler correctement les instructions du 65C02. On bascule entre les deux désassembleurs par Ctrl-Y, puis on utilise comme d'habitude la commande L du moniteur.*

Comparaison du contenu des ROMs de $D000 à $F7FF dans les APPLE :

II+	//e	//e+	//c	
D034- DA $F775-1	.. ..	.. ..	D034- DA $03F5-1	SHLOAD remplacé par &
D04E- DA $F3BC-1	.. ..	.. ..	D04E- DA $03F5-1	RECALL remplacé par &
D050- DA $F39F-1	.. ..	.. ..	D050- DA $03F5-1	STORE remplacé par &
D06C- DA $D8C9-1	.. ..	.. ..	D06C- DA $03F5-1	LOAD remplacé par &
D06E- DA $D8B0-1	.. ..	.. ..	D06E- DA $03F5-1	SAVE remplacé par &
D56D- LDA $200,X	... .	D56D- JSR $F78C	D56D- JSR $D8B5	nouvelle recherche des caractères en entrée
D5A8- LDA $200,X	... .	D5A8- JSR $F78C	D5A8- JSR $D8B5	
D5BE- LDA $201,X	... .	D5BE- JSR $F787	D5BE- JSR $D8B0	
D5E9- LDA $200,X	... .	D5E9- JSR $F78C	D5E9- JSR $D8B5	
D60B- LDA $200,X	... .	D60B- JSR $F79A	D60B- JSR $D8C3	
D6F9- JSR $ED24	... .	D6F9- JSR $F7AA	D6F9- JSR $D8D3	
D705- LDA $24	... ...	D705- JSR $F7B4	D705- JSR $D8DD	
D707- CMP #33	... ...	D708- NOP	D708- NOP	
D8B0- SEC	...	...	D8B0- LDA $201,X	routine SAVE remplacée par nouvelle routine de formatage des caractères en entrée
D8B1- LDA $AF	... ...	... ...		
D8B3- SBC $67	... ...	... ...	D8B3- BPL $D8C6	
D8B5- STA $50	... ...	... ...	D8B5- LDA $0E	
D8B7- LDA $B0	... ...	... ...	D8B7- BEQ $D8CF	
D8B9- SBC $68	... ...	... ...	D8B9- CMP #$22	'"' ?
D8BB- STA $51	... ...	... ...	D8BB- BEQ $D8CF	

```
D8BD- JSR $D8F0     ... .....    ... ....         D8BD- LDA $13
                                                  D8BF- CMP #$49          ctrl-c - $3A ?
D8C0- JSR $FECD     ... .....    ... .....        D8C1- BEQ $D8CD
D8C3- JSR $D901     ... .....    ... .....        D8C3- LDA $200,X
D8C6- JMP $FECD     ... .....    ... .....        D8C6- PHP
                                                  D8C7- CMP #$61          'a' ?
D8C9- JSR $D8F0     ... .....    ... .....        D8C9- BCC $D8CD
D8CC- JSR $FEFD     ... .....    ... .....        D8CB- AND #$5F          force en majuscule
                                                  D8CD- PLP
                                                  D8CE- RTS
D8CF- CLC           ... .....    ... .....        D8CF- LDA $200,X
D8D0- LDA $67       ... ...      ... ...          D8D2- RTS
D8D2- ADC $50       ... ...      ... ...          D8D3- PHA
D8D4- STA $69       ... ...      ... ...          D8D4- LDA #$20          affiche un espace
D8D6- LDA $68       ... ...      ... ...          D8D6- JSR $DB5C         devant le numéro de ligne
D8D8- ADC $51       ... ...      ... ...          D8D9- PLA
D8DA- STA $6A       ... ...      ... ...          D8DA- JMP $ED24
D8DC- LDA $52       ... ...      ... ...          D8DD- LDA $24
D8DE- STA $D6       ... ...      ... ...          D8DF- CMP #$21
D8E0- JSR $D901     ... .....    ... .....        D8E1- BIT $C01F
D8E3- JSR $FEFD     ... .....    ... .....        D8E4- BPL $D8EB         branche si 40 colonnes
D8E6- BIT $D6       ... ...      ... ...          D8E6- LDA $057B
D8E8- BPL $D8ED     ... .....    ... .....        D8E9- CMP #$49
D8EA- JMP $D665     ... .....    ... .....        D8EB- RTS
D8ED- JMP $D4F2     ... .....    ... .....        D8EC- LDA $C050
D8F0- LDA #$50      ... ....     ... ....         D8EF- JSR $D8F7
D8F2- LDY #$00      ... ....     ... ....         D8F2- LDA #$14
D8F4- STA $3C       ... ...      ... ...          D8F4- JMP $FB4B
D8F6- STY $3D       ... ...      ... ...          D8F7- LDY #$27
D8F8- LDA #$52      ... ....     ... ....         D8F9- STY $2D
D8FA- STA $3E       ... ...      ... ...          D8FB- JSR $F3CB
D8FC- STY $3F       ... ...      ... ...
D8FE- STY $D6       ... ...      ... ...          D8FE- LDA #$27
D900- RTS           ...          ...              D900- BCC $D903
D901- LDA $67       ... ...      ... ...          D902- ROL
D903- LDY $68       ... ...      ... ...          D903- TAY
D905- STA $3C       ... ...      ... ...          D904- LDA #$00
D907- STY $3D       ... ...      ... ...          D906- STA $30
D909- LDA $69       ... ...      ... ...          D908- JSR $F78B
D90B- LDY $6A       ... ...      ... ...          D90B- DEY
D90D- STA $3E       ... ...      ... ...          D90C- BPL $D904
D90F- STY $3F       ... ...      ... ...          D90E- RTS
D911- RTS           ...          ...              D90F- HEX 00,00,00 filler
--------
DADB- BEQ $DB16     ... .....    DADB- BEQ $DB19  *** *****
--------
DAE0- BEQ $DB16     ... .....    DAE0- BEQ $DB19  *** *****
--------
DB03- LDA $24       ... ...      DB03- JSR $F7B4  DB03- JSR $D8DD
DB05- CMP #$18      ... ....     DB06- BMI $DB11  *** *****
DB07- BCC $DB0E     ... .....    DB08- CMP #$18   *** ****        bug du TAB toujours là:
DB09- JSR $DAFB     ... .....    DB0A- BCC $DB11  *** *****       il faudrait CMP #$20
DB0C- BNE $DB2F     ... .....    DB0C- JSR $DAFB  *** *****
DB0E- ADC #$10      ... ....     DB0F- BNE $DB2F  *** *****
DB10- AND #$F0      ... ....     DB11- ADC #$10   *** ****
DB12- STA $24       ... ...      DB13- AND #$F0   *** ****
DB14- BCC $DB2F     ... .....    DB15- TAX        ***
DB16- PHP           ...          DB16- SEC        ***
DB17- JSR $E6F5     ... .....    DB17- BCS $DB25  *** *****
DB1A- CMP #$29      ... ....     DB19- PHP        ***
DB1C- BEQ $DB21     ... .....    DB1A- JSR $E6F5  *** *****
DB1E- JMP $DEC9     ... .....    DB1D- CMP #$29   *** ****
DB21- PLP           ...          DB1F- BNE $DB83  *** *****
DB22- BCC $DB2B     ... .....    DB21- PLP        ***
DB24- DEX           ...          DB22- BCC $DB2B  *** *****
DB25- TXA           ...          DB24- DEX        ***
DB26- SBC $24       ... ...      DB25- JSR $F7C3  DB25- JSR $F7CB
--------
DF02- JSR $F871     ... .....    ... .....        DF02- JSR $F7A6
--------
E006- HEX 00        ... ..       E006- HEX C4     E006- HEX 00    Pourquoi $C4 sur //e+ ?
                                                                  Crée des problèmes avec
                                                                  les programmes utilisant
```

```
F1EF- CPX #$30      ... ....     ... ....         F1EF- CPX #$50      $E006 comme identificateu
--------                                                              du Basic.
F1FD- CPX #$30      ... ....     ... ....         F1FD- CPX #$50
--------
F220- CPX #$30      ... ....     ... ....         F220- CPX #$50
--------
F228- TXA           ...          ...              F228- LDY $F0
F229- LDY $F0       ... ...      ... ...          F22A- JSR $F775
F22B- CPY #$28      ... ....     ... ....         F22D- TXA
F22D- BCS $F206     ... .....    ... .....        F22E- JMP $F39F
F22F- JMP $F800     ... .....    ... .....        F231- HEX 0D ??
--------
F235- TXA           ...          ...              F235- LDY $2C
F236- LDY $2C       ... ...      ... ...          F237- JSR $F775
F238- CPY #$28      ... ....     ... ....
F23A- BCS $F206     ... .....    ... .....        F23A- CPX #$30
F23C- LDY $F0       ... ...      ... ...          F23C- BCS $F206
F23E- JMP $F819     ... .....    ... .....        F23E- JMP $F796
--------
F246- CPY #$28      ... ....     ... ....         F246- JSR $F775
F248- BCS $F206     ... .....    ... .....        F249- LDA $F0
F24A- LDA $F0       ... ...      ... ...          F24B- JMP $F783
F24C- JMP $F828     ... .....    ... .....        F24E- HEX 00
--------
F396- JMP $FB40     ... .....    ... .....        F396- JMP $D8EC
--------
F39F- JSR $F7D9     ... .....    ... .....        F39F- LSR           remplace STORE et RECALL
                                                  F3A0- PHP
F3A2- LDY #$03      ... ....     ... ....         F3A1- JSR $F847
F3A4- LDA ($9B),Y   ... .......  ... .......      F3A4- PLP
F3A6- TAX           ...          ...              F3A5- LDA #$0F
F3A7- DEY           ...          ...              F3A7- BCC $F3AB
F3A8- LDA ($9B),Y   ... .......  ... .......      F3A9- ADC #$E0
F3AA- SBC #$01      ... ....     ... ....         F3AB- STA $2E
F3AC- BCS $F3AF     ... .....    ... .....        F3AD- PHY
F3AE- DEX           ...          ...              F3AE- JSR $F7BB
F3AF- STA $50       ... ...      ... ...          F3B1- BCC $F3BD
F3B1- STX $51       ... ...      ... ...          F3B3- PHX
F3B3- JSR $FECD     ... .....    ... .....        F3B4- LDA $30
F3B6- JSR $F7BC     ... .....    F3B6- JSR $F777  F3B6- TAX
                                                  F3B7- LSR
                                                  F3B8- TXA
F3B9- JMP $FECD     ... .....    ... .....        F3B9- ROR
                                                  F3BA- SEC
                                                  F3BB- STA $30
F3BC- JSR $F7D9     ... .....    ... .....        F3BD- JSR $F80E
F3BF- JSR $FEFD     ... .....    ... .....        F3C0- BCC $F3C9
F3C2- LDY #$02      ... ....     ... ....         F3C2- LDA $C054
F3C4- LDA ($9B),Y   ... .......  ... .......      F3C5- STX $30
F3C6- CMP $50       ... ...      ... ...          F3C7- PLX           dépile X
F3C8- INY           ...          ...              F3C8- CLC
F3C9- LDA ($9B),Y   ... .......  ... .......      F3C9- PLY           dépile Y
                                                  F3CA- RTS
F3CB- SBC $51       ... ...      ... ...          F3CB- LDA $C079
F3CD- BCS $F3D2     ... .....    ... .....        F3CE- EOR #$80
F3CF- JMP $D410     ... .....    ... .....        F3D0- AND $C018
F3D2- JSR $F7BC     ... .....    ... .....        F3D3- AND $C01F
F3D5- JMP $FEFD     ... .....    ... .....        F3D6- ASL
                                                  F3D7- RTS
carry=1 si mode 80 colonnes
--------
F775- LDA #$00      ... ....     F775- SEC        F775- JSR $F3CB     remplace SHLOAD
F777- STA $3D       ... ...      F776- HEX 90     F778- BCS $F77E
F779- STA $3F       ... ...      F777- CLC        F77A- CPY #$28
F77B- LDY #$50      ... ...      F778- STA $C007  F77C- BCS $F73F
F77D- STY $3C       ... ...      F77B- JSR $C500  F77E- CPY #$50
F77F- INY           ...          F77E- STA $C006  F780- BCS $F73F
F780- STY $3E       ... ...      F781- BCS $F784  F782- RTS
F782- JSR $FEFD     ... .....    F783- RTS        F783- PHA
F785- CLC           ...          F784- JMP $D410  F784- LDA $2D
F786- LDA $73       ... ...      F787- LDA $201,X F786- CMP #$30
F788- TAX           ...                           F788- PLA
F789- DEX           ...                           F789- BCS $F73F
```

F78A- STX $3E	... ...	F78A- BPL $F79D	F78B- PHA	
F78C- SBC $50	... ...	F78C- LDA $0E	F76C- JSR $F39F	
F78E- PHA	...			
F78F- LDA $74	... ...	F78E- BEQ $F7A6	F78F- PLA	
F791- TAY	...	F790- CMP #$22	F790- CMP $2D	
F792- INX	...	F792- BEQ $F7A6	F792- INA	incrémente A
F793- BNE $F796	... .	F794- LDA $13	F793- BCC $F78B	
F795- DEY	...		F795- RTS	
F796- STY $3F	... ...	F796- CMP #$49	F796- TXA	
F798- SBC $51	... ...	F798- BEQ $F7A6	F797- LDY $F0	
F79A- CMP $6E	... ...	F79A- LDA $200,X	F799- JSR $F39F	
F79C- BCC $F7A0	... .	F79D- PHP	F79C- CPY $2C	
F79E- BNE $F7A3	... .	F79E- CMP #$61	F79E- BCS $F795	
F7A0- JMP $D410	... .	F7A0- BCC $F7A4	F7A0- INY	
F7A3- STA $74	... ...	F7A2- AND #$5F	F7A1- JSR $F3AD	
		F7A4- PLP	F7A4- BRA $F79C	
F7A5- STA $70	... ...	F7A5- RTS		
F7A7- STA $3D	... ...	F7A6- LDA $200,X	F7A6- PHA	
F7A9- STA $E9	... ...	F7A9- RTS	F7A7- JSR $F7BB	
F7AB- PLA	...	F7AA- PHA	F7AA- PLA	
F7AC- STA $E8	... ...	F7AB- LDA #$20	F7AB- PHP	
F7AE- STA $73	... ...	F7AD- JSR $DB5C	F7AC- JSR $F871	
			F7AF- PLP	
F7B0- STA $6F	... ...	F7B0- PLA	F7B0- BCC $F7BA	
F7B2- STA $3C	... ...	F7B1- JMP $ED24	F7B2- STA $C054	
F7B4- JSR $FCFA	... .	F7B4- LDA $24	F7B5- CMP #$08	
F7B7- LDA #$03	... .	F7B6- CMP #$21	F7B7- ASL	
F7B9- JMP $FF02	... .	F7B8- BIT $C01F	F7B8- AND #$0F	
			F7BA- RTS	
F7BC- CLC	...	F7BB- BPL $F7C2	F7BB- JSR $F3CB	
F7BD- LDA $9B	... ...	F7BD- LDA $057B	F7BE- BCC $F7CA	
F7BF- ADC $50	... ...	F7C0- CMP #$49	F7C0- TYA	
F7C1- STA $3E	... ...	F7C2- RTS	F7C1- EOR #$01	
F7C3- LDA $9C	... ...	F7C3- TXA	F7C3- LSR	
F7C5- ADC $51	... ...	F7C4- BIT $C01F	F7C4- TAY	
			F7C5- BCC $F7CA	
F7C7- STA $3F	... ...	F7C7- BMI $F7D1	F7C7- LDA $C055	
F7C9- LDY #$04	... .	F7C9- HEX 2C	F7CA- RTS	
F7CB- LDA ($9B),Y	... .	F7CA- STA $24	F7CB- TXA	
F7CD- JSR $E0EF	... .	F7CC- SEC	F7CC- BIT $C01F	
		F7CD- TXA	F7CF- BMI $F7E3	
		F7CE- SBC $24		
F7D0- LDA $94	... ...	F7D0- RTS	F7D1- HEX 2C	
F7D2- STA $3C	... ...	F7D1- SBC $057B	F7D2- STA $24	
F7D4- LDA $95	... ...	F7D4- RTS	F7D4- SEC	
F7D6- STA $3D	... ...	F7D5- HEX 00,00	F7D5- TXA	
		F7D7- BRK	F7D6- SBC $24	
F7D8- RTS	...	F7D8- BRK	F7D8- RTS	

F7E0- LDA #$00	... .	... .	F7E0- STZ $14	
F7E2- STA $14	... ...	... ...	F7E2- RTS	
F7E4- JMP $D8F0	... .	... .	F7E3- SBC $057B	
			F7E6- RTS	

F7EB- TXA	...	F7EB- LDA #$28	*** ****	
F7EC- CMP #$28	... .	F7ED- CMP $21	*** ***	
F7EE- BCC $F7FA	... .	F7EF- BCS $F7F3	*** *****	
F7F0- SBC #$28	... .	F7F1- LDA $21	*** ***	
F7F2- PHA	...			
F7F3- JSR $DAFB	... .	F7F3- JSR $F7D2	*** *****	
F7F6- PLA	...	F7F6- STX $24	*** ***	
F7F7- JMP $F7EC	... .	F7F8- BCC $F7D0	F7F8- BCC $F7D8	
F7FA- STA $24	... ...	F7FA- TAX	***	
F7FC- RTS	...	F7FB- JSR $DAFB	*** *****	
F7FD- HEX CB,D2	... .	F7FE- BNE $F7EB	F7FE- BRA $F7EB	
F7FF- HEX D7	F7FF- HEX 78			contrôlé par la ROM interne Cx00

Comparaison du contenu des ROMs de $F800 à $FBFF dans les APPLE :

II+	//e	//e+	//c	
F890- BCC $F89B	... .	... .	F890- BCC $F897	

F893- BCS $F8A5	... .	... .	F893- BCS $F8A1	
F895- CMP #$A2	... .	... .	F895- AND #$87	
F897- BEQ $F8A5	... .	... .	F897- LSR	
F899- AND #$87	... .	... .	F898- TAX	
F89B- LSR	...	...	F899- LDA $F962,X	
F89C- TAX	...	...	F89C- JSR $F879	
F89D- LDA $F962,X	... .	... .	F89F- BNE $F8A5	
F8A0- JSR $F879	... .	... .	F8A1- LDY #$FC	les opcodes invalides sont désormais affichés 'NOP' et non plus '???'
F8A3- BNE $F8A9	... .	... .	F8A3- LDA #$00	
F8A5- LDY #$80	... .	... .	F8A5- TAX	
F8A7- LDA #$00	... .	... .	F8A6- LDA $F9A6,X	
F8A9- TAX	...	...	F8A9- STA $2E	
F8AA- LDA $F9A6,X	... .	... .	F8AB- AND #$03	
F8AD- STA $2E	... ...	... ...	F8AD- STA $2F	
F8AF- AND #$03	... .	F8AF- TAX	F8AF- JSR $FC35	
F8B1- STA $2F	... ...	F8B0- STY $2A	F8B2- BEQ $F8CC	
F8B3- TYA	...	F8B2- LDY #$10		
F8B4- AND #$8F	... .	F8B4- JMP $FBB4	F8B4-	
F8B6- TAX	...		F8B6- ...	
F8B7- TYA	...	F8B7- STA $C006	F8B7- ...	
F8B8- LDY #$03	... .		F8B8-	
F8BA- CPX #$8A	... .	F8BA- LDX #$02	F8BA-	
F8BC- BEQ $F8C9	... .	F8BC- LDA $C305,X	F8BC-	
F8BE- LSR	...		F8BE- ...	
F8BF- BCC $F8C9	... .	F8BF- CMP $FC9C,X	F8BF-	
F8C1- LSR	...		F8C1- ...	
F8C2- LSR	...	F8C2- BNE $F8CB	F8C2- ...	
F8C3- ORA #$20	... .	F8C4- DEX	F8C3-	
F8C5- DEY	...	F8C5- DEX	F8C5- ...	
F8C6- BNE $F8C2	... .	F8C6- BPL $F8BC	...	F8C8- INY
	...	F8C8- DEY	F8C8- ...	
F8C9- DEY	...	F8C9- BNE $F8BA	F8C9- ...	
F8CA- BNE $F8BE	... .	F8CB- STA $C007	F8CA-	
F8CC- RTS	...	F8CE- RTS	F8CC- ...	
F8CD- HEX FF,FF,FF	... ..,..,..	F8CF- BRK	F8CD- ... ..,..,..	

F918- LDA $F9B3,X	... .	... .	F918- LDA $F9B9,X	

F91E- LDA $F9B9,X	... .	... .	F91E- LDA $F9B3,X	

F962- HEX 04,20,54	... ..,..,..	... ..,..,..	F962- HEX 0F,22,FF	tables des opcodes
F965- HEX 30,0D,80	... ..,..,..	... ..,..,..	F965- HEX 33,CB,62	
F968- HEX 04,90	... ..,..	... ..,..	F968- HEX FF,73	

F96C- HEX 54	... ..	... ..	F96C- HEX FF	

F96E- HEX 0D,80,04	... ..,..,..	... ..,..,..	F96E- HEX CB,66,FF	
F971- HEX 90,04	... ..,..	... ..,..	F971- HEX 77,0F	

F974- HEX 54	... ..	... ..	F974- HEX FF	

F976- HEX 0D,80,04	... ..,..,..	... ..,..,..	F976- HEX CB,60,FF	
F979- HEX 90,04,20	... ..,..,..	... ..,..,..	F979- HEX 70,0F,22	
F97C- HEX 54,3B,0D	... ..,..,..	... ..,..,..	F97C- HEX FF,39,CB	
F97F- HEX 80,04,90	... ..,..,..	... ..,..,..	F97F- HEX 66,FF,7D	
F982- HEX 00	... ..	... ..	F982- HEX 0B	

F984- HEX 44	... ..	... ..	F984- HEX FF	

F986- HEX 0D,C8	... ..,..	... ..,..	F986- HEX CB,A6	
F988- HEX 44,00	... ..,..	... ..,..	F988- HEX FF,73	

F98C- HEX 44	... ..	... ..	F98C- HEX FF	

```
F98E- HEX 0D,C8      ... ..,..      ... ..,..      F98E- HEX CB,A6
F990- HEX 44,A9      ... ..,..      ... ..,..      F990- HEX FF,87
--------
F994- HEX 44         ... ..         ... ..         F994- HEX FF
--------
F996- HEX 0D,80      ... ..,..      ... ..,..      F996- HEX CB,60
F998- HEX 04,90      ... ..,..      ... ..,..      F998- HEX FF,70
--------
F99C- HEX 44         ... ..         ... ..         F99C- HEX FF
--------
F99E- HEX 0D,80      ... ..,..      ... ..,..      F99E- HEX CB,60
F9A0- HEX 04,90,26   ... ..,..,..   ... ..,..,..   F9A0- HEX FF,70,24
--------
F9A4- HEX 87,9A      ... ..,..      ... ..,..      F9A4- HEX 65,78
--------
F9AA- HEX 00,00,59   ... ..,..,..   ... ..,..,..   F9AA- HEX 59,4D,91
F9AD- HEX 4D,91,92   ... ..,..,..   ... ..,..,..   F9AD- HEX 92,86,4A
F9B0- HEX 86,4A      ... ..,..      ... ..,..      F9B0- HEX 85,9D
F9B2- HEX 85,9D      ... ..,..      ... ..,..      F9B2- HEX 49,5A
F9B4- HEX AC,A9,AC   ... ..,..,..   ... ..,..,..   F9B4- HEX D9,00,D8
F9B7- HEX A3,A8,A4   ... ..,..,..   ... ..,..,..   F9B7- HEX A4,A4,00
F9BA- HEX D9,00,D8   ... ..,..,..   ... ..,..,..   F9BA- HEX AC,A9,AC
F9BD- HEX A4,A4,00   ... ..,..,..   ... ..,..,..   F9BD- HEX A3,A8,A4
--------
F9D0- HEX 0          ... ..         ... ..         F9D0- HEX 1C
--------
F9E0- HEX 00         ... ..         ... ..         F9E0- HEX AD
--------
F9ED- HEX 00         ... ..         ... ..         F9ED- HEX 8A
--------
F9EF- HEX 00         ... ..         ... ..         F9EF- HEX 8B
--------
FA10- HEX 00         ... ..         ... ..         FA10- HEX C4
--------
FA20- HEX 0          ... ..         ... ..         FA20- HEX 06
--------
FA2D- HEX 00         ... ..         ... ..         FA2D- HEX 72
--------
FA2F- HEX 00         ... ..         ... ..         FA2F- HEX 72
--------
FA40- STA $45        ... ...        ... ...        FA40- PHA
FA42- PLA            ...            FA42- LDA $45  FA41- PLA
FA43- PHA            ...                           FA42- PLA
FA44- ASL            ...            FA44- JMP $C3FA FA43- JMP $C806
FA45- ASL            ...                           FA46- HEX FF
FA46- AS             ...                           FA47- STA $44
FA47- BMI $FA4C      ... .....      FA47- STA $C006 FA49- PLY
FA49- JMP ($03FE)    ... .......    FA4A- STA $45  FA4A- PLX
                                                   FA4B- PLA
FA4C- PLP            ...            ...            ...
FA4D- JSR $FF4C      ... .....      ... .....      FA4D- JSR $FF4A
--------
FA6F- LDA $C058      ... .....      ... .....      FA6F- JSR $C41C
FA72- LDA $C05A      ... .....      ... .....      FA72- JSR $CC04
FA75- LDA $C05D      FA75- LDY #$05 FA75- LDY #$09 FA75- STZ $04FF
FA78- LDA $C05F      FA77- JSR $FBB4 FA77- +++ +++++ FA78- LDA $C05F
                     FA7A- NOP      FA7A- +++
FA7B- LDA $CFFF      ... .....      ... .....      FA7B- JSR $FABD
--------
FA92- BNE $FAA3      ... .....      ... .....      FA92- BNE $FACF
--------
FA99- BNE $FAA3      ... .....      ... .....      FA99- BNE $FACF
--------
FAA3- JMP ($03F2)    ... .......    ... .......    FAA3- JSR $FF3A
FAA6- JSR $FB60      ... .....      ... .....      FAA6- JSR $FCCA
--------
FAB4- LDA #$C        ... ....       ... ....       FAB4- LDA #$C6     désormais on
                                                                       recherche un lecteur
FAB6- STX $00        ... ...        ... ...        FAB6- BRA $FB12    de disquette à
                                                                       partir du slot 6
FAB8- STA $01        ... ...        ... ...        FAB8- HEX 8A,8B    opcodes PHY, PLY
FABA- LDY #$07       ... ....       FABA- LDY #$05 FABA- HEX A5,AC,00 opcodes STZ, TRB, BRK
FABC- DEC $01        ... ...        ... ...        FABD- LDA #$FF
```

```
FABE- LDA $01        ... ...        ... ...        FABF- STA $04FB
FAC0- CMP #$C0       ... ....       ... ....
FAC2- BEQ $FA9B      ... .....      ... .....      FAC2- ASL $C062
FAC4- STA $07F8      ... .....      ... .....      FAC5- BIT $C061
FAC7- LDA ($00),Y    ... .......    ... .......    FAC8- BPL $FB2E
FAC9- CMP $FB01,Y    ... .......    ... .......    FACA- BCC $FAA3
FACC- BNE $FABA      ... .....      ... .....      FACC- JMP $C77C
FACE- DEY            ...            ...
FACF- DEY            ...            ...            FACF- JMP ($03F2)
FAD0- BPL $FAC7      ... .....      ... .....
FAD2- JMP ($0000     ... .......    ... .......    FAD2- HEX C1,D8,D9
FAD5- HEX EA,EA      ... ..,..      FAD5- HEX 00,00 FAD5- HEX D0,D3
--------
FAE9- LDA $FA1E,X    ... .......    ... .......    FAE9- LDA $F9D7,X
--------
FAF6- JSR $FDDA      ... .....      ... .....      FAF6- BRA $FB02
FAF9- INX            ...            ...            FAF8- HEX 74,74
FAFA- BMI $FAE4      ... .....      ... .....      FAFA- HEX 76,C6
FAFC- RTS            ...            ...            FAFC- HEX 00
--------
FB02- HEX 20,FF,00   ... ..,..,..   ... ..,..,..   FB02- JSR $FDDA
FB05- HEX F          ... ..         ... ..         FB05- INX
FB06- HEX 03,FF      ... ..,..      ... ..,..      FB06- BMI $FAE4
FB08- HEX 3C         ... ..         ... ..         FB08- RTS
FB09- APPLE ]        FB09- Apple ][ +++++ ++       +++++ ++
--------
FB12- HEX C2,C1      ... ..,..      ... ..,..      FB12- STX $00
FB14- HEX FF,C3      ... ..,..      ... ..,..      FB14- STA $01
FB16- HEX FF,FF,FF   ... ..,..,..   ... ..,..,..   FB16- JSR $FB60
FB19- HEX C1,D8,D9   ... ..,..,..   ... ..,..,..   FB19- JMP ($0000)
FB1C- HEX D0,D3      ... ..,..      ... ..,..      FB1C- HEX FF,FF
FB1E- LDA $C070      ... .....      ... .....      FB1E- JMP $C7DE
--------
FB4D- LDA #$00       ... ....       ... ....       FB4D- HEX EA,EA
FB4F- STA $20        ... ...        ... ...        FB4F- JSR $CE0A
FB51- LDA #$28       FB51- LDY #$08 FB51- LDY #$0C FB52- BRA $FB59
FB53- STA $21        FB53- BNE $FBB4 +++ +++++     FB54- ORA #$80
FB55- LDA #$18       ... ....       ... ....       FB56- JMP $CD54
FB57- STA $23        ... ...        ... ...
--------
FB63- LDY #$08       ... ....       FB63- LDY #$09 *** ****
FB65- LDA $FB08,Y    ... .......    FB65- LDA $FF09,Y FB65- LDA $FD02,Y
FB68- STA $040E,Y    ... .......    ... .......    FB68- STA $040D,Y
--------
FB94- JMP $FBFD      ... .....      ... .....      FB94- BIT $067B
FB97- SEC            ...            ...            FB97- BMI $FBFD
FB98- JMP $FC2C      ... .....      ... .....      FB99- BIT #$60
FB9B- TAY            ...            ...            FB9B- JSR $FB54
FB9C- LDA $FA48,Y    ... .......    ... .......    FB9D- JSR $C3B8
FB9F- JSR $FB97      ... .....      ... .....      FBA0- INC $057B
FBA2- JSR $FD0C      FBA2- JSR $FD21 +++ +++++     FBA3- LDA $057B
FBA5- CMP #$CE       ... ....       ... ....       FBA6- BIT $C01F
FBA7- BCS $FB97      ... .....      ... .....
FBA9- CMP #$C9       ... ....       ... ....       FBA9- BMI $FBB0
FBAB- BCC $FB97      ... .....      ... .....      FBAB- STA $047B
FBAD- CMP #$CC       ... ....       ... ....       FBAE- STA $24
FBAF- BEQ $FB97      ... .....      ... .....      FBB0- BRA $FBF8
FBB1- BNE $FB9B      ... .....      ... .....      FBB2- HEX FF
FBB3- HEX EA         FBB3- HEX 06   +++ ++         +++ ++                version
FBB4- NOP            FBB4- PHP      FBB4- BIT $C015 FBB4- BPL $FBBC
FBB5- NOP            FBB5- SEI
FBB6- HEX EA,EA,EA   FBB6- BIT $C015 FBB7- PHP     FBB6- CMP #$A0
FBB9- NOP            FBB9- PHP      FBB8- STA $C007 FBB8- BCC $FBBC
FBBA- HEX EA,EA,EA   FBBA- STA $C007 FBBB- JMP $C100 FBBA- AND $32
FBBD- HEX EA,EA,EA   FBBD- JMP $C100 FBBE- BRK     FBBC- JMP $FDF6
                     ...            FBBF- BRK      FBBF- HEX FF
FBC0- HEX EA         ... ..         FBC0- HEX E0   FBC0- HEX 00
```

Comparaison du contenu des ROMs de $FC00 à $FFFF dans les APPLE :

II+	//e	//e+	//c	
FC06- BEQ $FC62	... .	... .	FC06- BEQ $FC73	

FC10- DEC $24	... ...	... ...	FC10- JSR $FEE2	
FC12- BPL $FBFC	... .	... .	FC13- BPL $FBFC	
FC14- LDA $21	... ...	... ...	FC15- LDA $21	
FC16- STA $24	... ...	... ...	FC17- JSR $FEEB	
FC18- DEC $24	... ...	... ...		

FC1E- BCS $FC2B	... .	FC1E- BCS $FBFC	*** *****	

FC22- LDA $25	... ...	... ...	FC22- BRA $FC86	
FC24- JSR $FBC1	... .	FC24- STA $28	FC24-	
		FC26- TYA		
FC27- ADC $20	... ...	FC27- LDY #$04	FC27- LDA $20	
FC29- STA $28	... ...	FC29- BNE $FBB4	FC29- BIT $C01F	
FC2B- RTS	...	FC2B- HEX 00	FC2C- BPL $FC30	
FC2C- EOR #$C0	... .	... .	FC2E- LSR	
FC2E- BEQ $FC58	... .	... .	FC2F- CLC	
FC30- ADC #$FD	... .	... .	FC30- ADC $28	
FC32- BCC $FBF4	... .	... .	FC32- STA $28	
FC34- BEQ $FC10	... .	... .	FC34- RTS	
			FC35- TYA	
FC36- ADC #$FD	... .	... .	FC36- LDX #$16	
FC38- BCC $FC66	... .	... .	FC38- CMP $FEFE,X	
FC3A- BEQ $FC1A	... .	... .	FC3B- BEQ $FC80	
FC3C- ADC #$FD	... .	... .	FC3D- DEX	
FC3E- BCC $FC9C	... .	... .	FC3E- BPL $FC38	
FC40- BNE $FC2B	... .	FC40- BNE $FBFC	FC40- RTS	
			FC41- HEX FF	
FC42- LDY $24	FC42- LDY #$00	FC42- LDY #$0A	FC42- BRA $FC5D	
FC44- LDA $25	FC44- BEQ $FC72	FC44- BNE $FC29	FC44-	
FC46- PHA	FC46- (C) 1981-	FC46- BIT $C01F	FC46- ...	
FC47- JSR $FC24		FC49- BPL $FC4F	FC47-	
FC4A- JSR $FC9E		FC4B- LDY #$00	FC4A-	
FC4D- LDY #$00		FC4D- BEQ $FC5A	FC4D-	
FC4F- PLA	FC4F- 82, APPLE	FC4F- TYA	FC4F- ...	
FC50- ADC #$00		FC50- PHA	FC50- INA	
FC52- CMP $23		FC51- JSR $FB78	FC51- CMP $23	
FC54- BCC $FC46		FC54- PLA	FC53- BCC $FC46	
FC56- BCS $FC22		FC55- LDY $35	FC55- BCS $FC22	
		FC57- RTS	FC57- HEX FF	
FC58- LDA $22	FC58- LDY #$01	FC58- LDY #$05	FC58- JSR $CDA5	
FC5A- STA $25	FC5A- BNE $FC72	FC5A- JMP $FBB4	FC5B- BRA $FC44	
FC5C- LDY #$00	FC5C- RICK A	FC5D- HEX 53	FC5D- JSR $CC9D	"RICK"-Auricchio
FC5E- STY $24		FC5E- HEX 5C,5C		
FC60- BEQ $FC46		FC60- HEX 00,00	FC60- BRA $FC44	
FC62- LDA #$00	... .	... .	FC62- BRA $FC73	
FC64- STA $24	... ...	... ...	FC64- PLX,PLX ???	

FC6C- BCC $FC24	... .	... .	FC6C- BCC $FC88	

FC70- LDA $22	FC70- LDY #$02	FC70- LDY #$06	FC70- JMP $CB35	
FC72- PHA	FC72- JMP $FBB4	FC72- BNE $FC29		
FC73- JSR $FC24	FC75- PHA	FC74- STA $C006	FC73- JSR $FEE9	
FC76- LDA $28	FC76- LDA $C018	FC77- JMP ($03FE)	FC76- BIT $04FB	
FC78- STA $2A	FC79- ASL		FC79- BPL $FC85	branche si Pascal
FC7A- LDA $29	FC7A- PLA	+++		
FC7C- STA $2B	FC7B- BIT $C01C	FC7B- STA $07F8	FC7B- JSR $FD44	
FC7E- LDY $21	FC7E- PHP	FC7E- CMP #$C1	FC7E- BRA $FC66	
FC80- DEY		FC80- BCC $FC8F	FC80- LDA $FF15,X	
FC81- PLA	FC81- STA $C054			
FC82- ADC #$01		FC82- STA $CFFF	FC83- LDY #$00	
FC84- CMP $23	FC84- BIT $C015	FC85- LDY #$00	FC85- RTS	
FC86- BCS $FC95	FC87- STA $C006	FC87- LDX $01	FC86- LDA $25	
FC88- PHA	FC8A- CLI	FC89- STA $01	FC88- STA $05FB	
FC89- JSR $FC24	FC8B- SEI	FC8B- LDA ($00),Y	FC8B- BRA $FC24	
FC8C- LDA ($28),Y	FC8C- BPL $FC91	FC8D- STX $01	FC8D- JSR $CC9D	nouveau CLREOL
FC8E- STA ($2A),Y	FC8E- STA $C007	FC8F- STA $C007	FC90- LDA #" "	nouveau CLEOLZ

FC90- DEY	FC91- PLP			
FC91- BPL $FC8C	FC92- BCC $FC99	FC92- JMP $C47C	FC92- BIT $067B	
FC93- BMI $FC76				
FC95- LDY #$00	FC94- BPL $FC99	FC95- BCC $FC99	FC95- BMI $FC99	
FC97- JSR $FC9E	FC96- BIT $C055	FC97- AND $32	*** ***	
	FC99- RTS			
FC9A- BCS $FC22	FC9A- HEX EA,EA	FC99- JMP $FDF7	FC99- JMP $CBC2	
FC9C- LDY $24	FC9C- CLC	FC9C- SEC	FC9C- BRA $FC8D	
	FC9D- HEX B0	FC9D- HEX 90		
FC9E- LDA #$A0	FC9E- SEC	FC9E- CLC	FC9E- BRA $FC90	
FCA0- STA ($28),Y	FC9F- STY $1F	FC9F- STY $2A	FCA0- LDY #$00	$1F était une adresse en page zéro inutilisé sur II+
FCA2- INY	FCA1- LDY #$03	FCA1- LDY #$07	FCA2- BRA $FC90	
FCA3- CPY #$21	FCA3- BCC $FC72	FCA3- BCS $FD1D	FCA4- JMP ($CD2A,X)	
FCA5- BCC $FCA0	FCA5- INY	+++		
FCA7- RTS	FCA6- BNE $FC72	FCA6- BNE $FD1D	FCA7- HEX FF	

FCC9- LDY #$4B	... .	FCC9- STA $C007	FCC9- RTS	fonction cassette supprimée
			FCCA- LDY #$B0	
FCCB- JSR $FCDB	... .	FCCC- JSR $C567	FCCC- STZ $3C	
FCCE- BNE $FCC9	... .	FCCF- JMP $FEC5	FCCE- LDX #$BF	
FCD0- ADC #$FE	... .		FCD0- STA $3D	
FCD2- BCS $FCC9	... .	FCD2- STA $C006	FCD2- LDA #$A0	
FCD4- LDY #$21	... .	FCD5- JSR $F94A	FCD4- STA ($3C),Y	
FCD6- JSR $FCDB	... .	FCD8- LDA #$DE	FCD6- DEY	
			FCD7- STA ($3C),Y	
FCD9- INY	...		FCD9- DEX	
FCDA- INY	...	FCDA- JSR $FDED	FCDA- CPX #$01	
FCDB- DEY	...			
FCDC- BNE $FCDB	... .		FCDC- BNE $FCD0	
FCDE- BCC $FCE5	... .	FCDD- JSR $FF3A	FCDE- STA $C001	
FCE0- LDY #$32	... .	FCE0- JMP $FCF0	FCE1- LDA $C055	
FCE2- DEY	...		FCE4- SEC	
FCE3- BNE $FCE2	... .	FCE3- STA $C006	FCE5- LDX #$88	
FCE5- LDY $C020	... .	FCE6- JSR $F8D0	FCE7- LDA $CB27,X	
FCE8- LDY #$2C	... .			
FCEA- DEX	...	FCE9- JSR $F953	FCEA- BCC $FCF6	
FCEB- RTS	...			
FCEC- LDY #$08	... .	FCEC- STY $3B	FCEC- CMP $0477,X	
FCEE- PHA	...	FCEE- STA $3A	FCEF- CLC	
FCEF- JSR $FCFA	... .	FCF0- LDA #$A1	FCF0- BNE $FCF6	
FCF2- PLA	...	FCF2- STA $33	FCF2- CPX #$82	
FCF3- ROL	...			
FCF4- LDY #$3A	... .	FCF4- JSR $FD67	FCF4- BCC $FCFC	
FCF6- DEX	...		FCF6- STA $0477,X	initialise pour 80 colonnes
FCF7- BNE $FCEE	... .	FCF7- STA $C007	FCF9- DEX	
FCF9- RTS	...			
FCFA- JSR $FCFD	... .	FCFA- JMP $CF9C	FCFA- BNE $FCE7	
FCFD- DEY	...	FCFD- LDA $0200,Y	FCFC- LDA $C054	
FCFE- LDA $C060	... .	FD00- INY	FCFF- STA $C000	
FD01- EOR $2F	... ...	FD01- CMP #$E1	FD02- RTS	
FD03- BPL $FCFD	... .	FD03- BCC $FD0B	FD03- Apple //c	
FD05- EOR $2F	... ...	FD05- CMP #$FB		
FD07- STA $2F	... ...	FD07- BCS $FD0B		
FD09- CPY #$80	... .	FD09- AND #$DF		
FD0B- RTS	...	...		
FD0C- LDY $24	... ...	FD0C- LDY #$0B	FD0C-	
FD0E- LDA ($28),Y	... .	FD0E- BNE $FD13	FD0E-	
FD10- PHA	...	FD10- JMP $FD18	FD10- HEX EA,EA,EA	
FD11- AND #$3F	... .			
FD13- ORA #$40	... .	FD13- JSR $FBB4	FD13- HEX EA,EA,EA	
FD15- STA ($28),Y	... .	FD16- HEX EA,EA	*** **,**	
FD17- PLA	...			

FD1B- INC $4E	FD1B- LDY #$06	FD1B- LDY #$03	FD1B- STA ($28),Y	
FD1D- BNE $FD21	FD1D- JMP $FBB4	+++ ++++	FD1D- JSR $CC4C	
FD1F- INC $4F	FD20- NOP	+++	FD20- JSR $CC70	
FD21- BIT $C000	FD21- JSR $FD0C	+++ +++	FD23- BPL $FD20	
FD24- BPL $FD1B	FD24- LDY #$07	FD24- LDY #$01	FD25- PHA	
FD26- STA ($28),Y	FD26- JMP $FBB4	FD26- BNE $FD1D	FD26- LDA #$8	
FD28- LDA $C000	FD29- STA $C006	FD28- LSR $07F8	FD28- BIT $04FB	

FD2B- BIT $C010	FD2C- PLP	FD2B- JMP $FD0C	FD2B- BNE $FD4A
	FD2D- RTS		FD2D- PLA
FD2E- RTS	...	FD2E- NOP	FD2E- CMP #$9B
FD2F- JSR $FD0C	FD2F- JSR $FD21	+++ +++++	FD30- BNE $FD38
FD32- JSR $FBA5	... .	... .	FD32- JMP $CCCC
FD35- JSR $FD0C	... .	FD35- JSR $FD28	FD35- JMP $CCED
FD38- CMP #$9B	... .	... .	FD38- BIT $067B
FD3A- BEQ $FD2F	... .	... .	FD3B- BMI $FD44
FD3C- RTS	...	...	
FD3D- LDA $32	... ...	FD3D- LDY #$0F	FD3D- CMP #$95
FD3F- PHA	...	FD3F- JSR $FBB4	FD3F- BNE $FD44
FD40- LDA #$F	... .	FD41- JSR $CC1D	
FD42- STA $32	FD42- HEX EA,EA	FD42- LDY $24	FD44- PHA
FD44- LDA $0200,X	... .	FD44- STA $0200,X	FD45- LDA #$08
FD47- JSR $FDED	... .	... .	FD47- TSB $04FB
FD4A- PLA	...	...	FD4A- PLA
FD4B- STA $32	... ...	FD4B- HEX EA,EA	FD4B- RTS
			FD4C- NOP
FD4D- LDA $0200,X	... .	... .	FD4D- JSR $C3A6

FD64- JSR $FDED	... .	... .	FD64- JSR $C3A6

FD75- JSR $FD35	.. ..	... .	FD75- JSR $CCED

FD7A- BNE $FD7E	... .	FD7A- BNE $FD84	*** *****
FD7C- LDA ($28),Y	... .	... .	FD7C- JSR $CC1D
FD7E- CMP #$E0	... .	FD7E- BIT $C01F	FD7F- HEX EA,EA
FD80- BCC $FD84	... .	FD81- BMI $FD3D	
FD82- AND #$DF	FD82- AND #$FF	FD83- NOP	FD81- HEX EA,EA,EA

FD89- BNE $FD3D	... .	FD89- BNE $FD47	FD89- BNE $FD4D

FDF0- CMP #$A0	... .	FDF0- PHA	FDF0- BIT $067B
FDF2- BCC $FDF6	... .	FDF1- CMP #$A0	
FDF4- AND $32	... ...	FDF3- JMP $FC95	FDF3- JMP $FBB4
FDF6- STY $35	... ...	FDF6- PHA	FDF6- ... ...
FDF8- PHA	...	FDF7- STY $35	FDF8- ...
FDF9- JSR $FB78	... .	FDF9- TAY	FDF9- ... .
FDFC- PLA	...	FDFA- PLA	FDFC- ...
FDFD- LDY $35	... ...	FDFB- JMP $FC46	FDFD- ... ...
FDFF- RTS	...	FDFE- HEX EA,EA	FDFF- ...

FE9F- BEQ $FEA7	... .	FE9F- BEQ $FEA5	FE9F- BNE $FEA7
FEA1- ORA #$C0	... .	... .	FEA1- CPY #$1B
FEA3- LDY #$00	... .	... .	FEA3- BEQ $FEDE
FEA5- BEQ $FEA9	... .	FEA5- STY $00,X	FEA5- BRA $FEC2
FEA7- LDA #$FD	... .	FEA7- STA $01,X	FEA7- ORA #$C0
FEA9- STY $00,X	... .	FEA9- LDY #$0E	FEA9- LDY #$00
FEAB- STA $01,X	... .	FEAB- JMP $FBB4	FEAB- STY $00,X
FEAD- RTS	...		FEAD- STA $01,X
FEAE- HEX EA,EA	FEAE- HEX EA,D1	FEAE- HEX EA,00	FEAF- RTS

FEC2- RT	...	...	FEC2- DEA
FEC3- NOP	...	...	FEC3- STA $07FB
FEC4- RTS	...	...	
FEC5- HEX EA,EA,EA	FEC5- Bryan	FEC5- STA $C006	FEC6- LDA #$F7
FEC8- HEX EA,EA		FEC8- RTS	FEC8- BRA $FECE
		FEC9- RTS	

FECD- LDA #$40	... .	... .	FECD- RTS
FECF- JSR $FCC9	... .	FECF- STA $C007	FECE- STA $067B
FED2- LDY #$27	... .	ED2- JSR $C5AA	FED1- STA $C00E
FED4- LDX #$00	... .	FED5- BEQ $FF03	FED4- TSB $04FB
FED6- EOR ($3C,X)	... .		FED7- PHX
FED8- PHA	...	FED7- LDY #$1	FED8- PHY
FED9- LDA ($3C,X)	... .	FED9- LDA $43	FED9- JSR $CDCD
FEDB- JSR $FEED	... .	FEDB- BEQ $FEE1	FEDC- PLY
		FEDD- CMP ($3C),Y	FEDD- PLX
FEDE- JSR $FCBA	... .	FEDF- BNE $FEEB	FEDE- LDA #$FD
FEE1- LDY #$1D	... .	FEE1- DEY	FEE0- BRA $FEAB

FEE3- PLA	...	FEE2- LDA $42	FEE2- PHY
FEE4- BCC $FED4	... .	FEE4- CMP ($3C),Y	FEE3- JSR $CC9D
FEE6- LDY #$22	... .	FEE6- BNE $FEEB	FEE6- DEY
FEE8- JSR $FEED	... .	FEE8- JSR $FD92	FEE7- BRA $FEEE
FEEB- BEQ $FF3A	... .	FEEB- JSR $FCBA	FEE9- LDA #$01
FEED- LDX #$10	... .	FEEE- BCC $FED7	FEEB- DEA
FEEF- ASL	...		FEEC- PHY
			FEED- TAY
		FEF0- RTS	FEEE- JSR $CCAD
FEF0- JSR $FCD6	... .	FEF1- LDY #$0D	FEF1- PLY
FEF3- BNE $FEEF	... .	FEF3- JSR $FBB4	FEF2- LDA $057B
FEF5- RTS	...		FEF5- ...

FEFD- JSR $FCFA	... .	FEFD- STA $C007	FEFD- RTS
FF00- LDA #$16	... .	FF00- JSR $C5D1	FEFE- HEX 12,14,1A
FF02- JSR $FCC9	... .	FF03- STA $C006	FF01- HEX 1C,32,34
FF05- STA $2E	... ...	FF06- BEQ $FF3A	FF04- HEX 3A,3C,52
FF07- JSR $FCFA	... .	FF08- BNE $FF2D	FF07- HEX 5A,64,72
FF0A- LDY #$24	... .	FF0A- Apple //e	FF0A- HEX 74,7A,7C
FF0C- JSR $FCFD	... .		FF0D- HEX 89,92
FF0F- BCS $FF0A	... .		FF0F- HEX 9C,9E
FF11- JSR $FCFD	... .		FF11- HEX B2,D2,F2
FF14- LDY #$3B	... .	FF13- JSR $FCFD	FF14- HEX FC,38
FF16- JSR $FCEC	... .	FF16- CMP #$A0	FF16- HEX FB,37,FB
FF19- STA ($3C,X)	... .	FF18- BEQ $FF13	
		FF1A- RTS	FF19- HEX 39,21
FF1B- EOR $2E	... ...	FF1B- BCS $FF8A	FF1B- KEX 36,21
FF1D- STA $2E	... ...	FF1D- CMP #$A0	FF1D- HEX 3A,F8
FF1F- JSR $FCBA	... .	FF1F- BNE $FF49	FF1F- HEX FA,3B,FA
FF22- LDY #$35	... .	FF21- LDA $0200,Y	FF22- HEX F9,22
FF24- BCC $FF16	... .	FF24- LDY #$07	FF24- HEX 21,3C
FF26- JSR $FCEC	... .	FF26- CMP #$8D	FF26- HEX FA,FA,3D
FF29- CMP $2E	... ...	FF28- BEQ $FFA7	FF29- HEX 3E,3F
		FF2A- INY	
FF2B- BEQ $FF3A	... .	FF2B- BEQ $FF69	FF2B- HEX FC,00

FF78- LDY #$17	... .	... .	FF78- LDY #$13

FF7D- CMP $FFCC,Y	... .	... .	FF7D- CMP $FFCD,Y

FFAD- LDA $0200,Y	... .	FFAD- JSR $FCFD	FFAD- ... .
FFB0- INY	...	FFB0- NOP	FFB0- ...
FFB1- EOR #$B0	... .	... .	FFB1- JSR $C399
FFB3- CMP #$0A	... .	... .	FFB4- EOR #$B0
FFB5- BCC $FF8A	... .	... .	FFB6- CMP #$0A
FFB7- ADC #$88	... .	... .	FFB8- BCC $FF8A
FFB9- CMP #$FA	... .	... .	FFBA- BRA $FFF3
FFBB- BCS $FF8A	... .	FFBB- JMP $FF1B	FFBC- HEX FF,41 ??
FFBD- RTS	RTS		

FFC1- LDA $FFE3,Y	... .	... .	FFC1- LDA $FFE0,Y

FFCC- HEX BC,B2,BE	... ..,..,..	... ..,..,..	FFCC- HEX EA,BC,B2
FFCF- HEX B2,EF,C4	... ..,..,..	FFCF- HEX 9A,EF,C4	FFCF- HEX BE,EF,C4
FFD2- HEX B2,A9,BB	... ..,..,..	FFD2- HEX BC,A9,BB	FFD2- HEX A9,BB,A6
FFD5- HEX A6,A4,06	... ..,..,..	... ..,..,..	FFD5- HEX A4,06,95
FFD8- HEX 95,07,02	... ..,..,..	... ..,..,..	FFD8- HEX 07,02,05
FFDB- HEX 05,F0,00	... ..,..,..	... ..,..,..	FFDB- HEX 00,93,A7
FFDE- HEX EB,93,A7	... ..,..,..	... ..,..,..	FFDE- HEX C6,99,B2
FFE1- HEX C6,99,B2	... ..,..,..	... ..,..,..	FFE1- HEX C9,BE,35
FFE4- HEX C9,BE,C1	... ..,..,..	FFE4- HEX C9,BE,F0	FFE4- HEX 8C,96,AF
FFE7- HEX 35,8C,C4	... ..,..,..	FFE7- HEX 35,8C,D6	FFE7- HEX 17,17,2B
FFEA- HEX 96,AF,17	... ..,..,..	... ..,..,..	FFEA- HEX 1F,83,7F
FFED- HEX 17,2B,1F	... ..,..,..	... ..,..,..	FFED- HEX 5D,B5,17
FFF0- HEX 83,7F,5D	... ..,..,..	... ..,..,..	FFF0- HEX 17,F5,03
FFF3- HEX CC,B5	... ..,..	... ..,..	FFF3- ADC #$88
FFF5- HEX FC,17	... ..,..	... ..,..	FFF5- CMP #$FA
FFF7- HEX 17,F5	... ..,..	... ..,..	FFF7- BCS $FF8A
FFF9- HEX 03	... ..	... ..	FFF9- RTS

FFFE- HEX 40,FA	... ..,..	FFFE- HEX FA,C3	FFFE- HEX 03,C8

PLE sur l'Apple IIc

Yvan Koenig

Vous venez de troquer votre Apple II+ ou IIe contre le portable IIc. Les caractères souris perturbent un peu l'affichage des programmes que vous utilisiez régulièrement sur l'ancien Apple.

Outre ce problème, le célèbre Program Line Editor (PLE), entre autres, ne fonctionne plus : lors de l'édition des lignes par Ctrl-E, un message d'erreur s'ajoute après la dernière instruction.

Le patch suivant, incorporé au programme Basic qui lance PLE, résoud le problème :

```
5 YK = PEEK (175) + 256
  * PEEK (176) - 1196
6 IF PEEK (64448) = 0
  THEN POKE YK, 170 :
  POKE YK + 4, 254
7 IF PEEK (64448) = 224
  THEN POKE YK, 164 :
  POKE YK + 4, 254
```

La ligne 5 détermine l'adresse du patch. La ligne 6 s'exécute si on se trouve effectivement sur un IIc. La ligne 7 est d'un usage plus personnel : j'ai modifié la ROM du kit de transformation IIe en IIc afin de bénéficier du mini-assembleur, d'un désassembleur qui connaît les nouveaux codes du 65C02 et de la basse résolution 80 colonnes ; tout cela au prix de la perte de l'auto-test. Les lecteurs peuvent me contacter à ce sujet à l'adresse suivante :

Yvan Koenig - 4, rue du Stade 06220 Vallauris

Défilement graphique... Défilement graphique... Défilement graphique... D

Patrick Quettier

Cette routine, en langage machine, est destinée à être intégrée dans un programme principal (en Basic) afin de faire défiler un texte sur l'écran, dans des conditions peu courantes.
Une chaîne alphanumérique (de 255 caractères au maximum) étant définie par le programme Basic, cette routine transforme les caractères de la chaîne en lettres géantes (1 caractère = 16 lignes de 12 colonnes) dans la page graphique haute résolution 1. Cette visualisation s'accompagne d'un défilement de tous les caractères, depuis le premier caractère apparaissant à droite de l'écran jusqu'au dernier disparaissant à gauche.

Utilisation dans un programme Basic

Précautions

Pour éviter tout conflit de mémoire entre le Basic, la page graphique haute résolution 1 et la routine en langage machine, il est souhaitable de charger le Basic (par exemple) au-dessus de la page graphique : c'est-à-dire à partir de 16384 ($4000). Sous DOS 3.3, cela laisse environ 21 Ko de libre pour le Basic et ses variables. On peut obtenir ce résultat de deux façons :

- il faut entrer au clavier
  POKE 103,1
  POKE 104,64
  POKE 16384,0 puis RUNer le programme Basic ;
- ou alors, il faut écrire un petit programme Basic qui lancera le programme Basic principal. On peut prendre comme exemple le programme de démonstration DEFILE.DEMO.LANCE qui lance le programme principal DEFILE.DEMO.PROG. Il suffit de le reproduire en mettant le nom de son propre programme Basic à la ligne 100.

Mode d'emploi

Le programme Basic devra, dès le début, contenir la ligne suivante (les numéros de lignes sont donnés à titre indicatif) :

```
100 PRINT CHR$(4); "BLOAD
PARAMETRES+DEFILE.OBJ0"
```

L'objet de cette ligne est de charger en mémoire la routine en langage machine.

Le défilement du texte se faisant en page graphique haute résolution 1, il peut être intéressant d'afficher un dessin (fixe) en plus du texte qui défile. Il faut alors charger en mémoire une page graphique déjà existante sur la disquette (pour cela ajoutez la ligne suivante) :

```
200 PRINT CHR$(4); "BLOAD
nom page graphique,A8192"
```

Il est aussi possible de réaliser le dessin à l'aide des instructions graphiques du Basic (HPLOT, etc...).

Il faut ensuite régler la tabulation verticale de la phrase qui va défiler : la page haute résolution contient 192 lignes (numérotées de 0 à 191), les caractères utilisent 16 lignes. Il faut ajouter :

```
210 POKE 251,V
```

avec V compris entre 0 (haut de l'écran) et 176 (bas de l'écran).

La chaîne alphanumérique qui va défiler doit être mise dans la variable IP$:

```
220 IP$="...phrase..."
ou
220 INPUT IP$
```

Tous les caractères de code ASCII 32 à 127 sont acceptés : cela correspond au jeu de 96 caractères de l'Apple //e (ou //c) en mode AZERTY.

Si la page graphique n'est pas encore affichée à l'écran, il faut la commuter : plusieurs méthodes sont possibles, en voici une qui n'efface pas le contenu de la page et l'affiche dans son intégralité (192 lignes).

```
230 POKE 49239,0:POKE 492
36,0:POKE 49234,0:POKE 492
32,0
```

Le lancement de la routine se fait par :

```
240 CALL 4736
```

Cette instruction fait défiler la chaîne IP$ une seule fois, mais le programme Basic peut la renouveler à plusieurs reprises (la tabulation et la chaîne IP$ ne sont pas modifiées).

En résumé

Voici, comme exemple, un (mini) programme qui utilise la routine :

```
100 PRINT CHR$(4); "BLOAD
    PARAMETRES+DEFILE.
    OBJ0"
200 PRINT CHR$(4); "BLOAD
    DEFILE.DEMO.PIC,A81
    92"
210 POKE 251,160
220 INPUT IP$
230 POKE 49239,0:POKE
    49236,0:POKE 49234,0:
    POKE 49232,0
240 CALL 4736
```

Modifications possibles

Il est possible de modifier la vitesse de défilement en ajoutant :
235 POKE 4960,Y: POKE 4962,

X avec X et Y compris entre 1 et 255. Actuellement, Y=15 et X=255. On obtient aussi de bons résultats avec Y=4 et X=1. Cela revient à modifier les lignes 187 et 188 du fichier source DEFILE.

N.D.L.R. : la vitesse diminue avec X (poids faible) et Y (poids fort); (noter que 0=256).

Il est possible de modifier le fichier PARAMETRES+DEFILE .OBJ0 pour obtenir l'affichage des caractères en mode inverse. Après chargement de ce fichier, il faut faire :

110 FOR T=2432 TO 4735: POKE T,255 - PEEK(T):NEXT

Remarque : les effets de deux exécutions successives de cette ligne s'annulent.

On peut aussi choisir de loger la routine à une autre adresse. Pour cela, il faut assembler le fichier DEFILE après avoir mis à la ligne 18 la nouvelle valeur choisie pour DB, puis créer un nouveau fichier sur une disquette :

BLOAD DEFILE.OBJ0 (nouveau)
BLOAD PARAMETRES,AX avec X=DB-2688
BSAVE PARAMETRES+ DEFILE.OBJ0, AX, L2922

Détails des programmes

DEFILE est le fichier source en assembleur ToolKit.

DEFILE.OBJ0 est le fichier objet de DEFILE.
adresse de début : 4736 ($1280)
adresse de fin : 4975 ($136F)
longueur : 240 octets

PARAMETRES est le bloc de paramètres utilisé par DEFILE. OBJ0.
adresse de début : 2048 ($0800)
adresse de fin : 4735 ($127F)
longueur : 2688 octets

PARAMETRES+DEFILE.OBJ0 réunit les 2 fichiers précédents.
adresse de début : 2048 ($0800)
adresse de fin : 4975 ($136F)
longueur : 2928 octets

Remarque : c'est le seul fichier qui soit nécessaire pour utiliser la routine à partir d'un programme Basic (dans les conditions standard).

DEFILE.DEMO.PROG est un programme de démonstration en Basic (attention, il ne faut pas le RUNer directement !).

DEFILE.DEMO.LANCE est le petit programme Basic qu'il faut RUNer pour obtenir une démonstration.

SHIFT+ONERR : routines utilisées par le programme de démonstration.

DEFILE.DEMO.PIC : page graphique haute résolution utilisée par le programme de démonstration *(ne figure pas sur la disquette).*

Programme DEFILE.DEMO.LANCE

```
10 REM ********************************
20 REM *                              *
30 REM *  Animation graphique - Démo  *
40 REM *                              *
50 REM *            Lancement         *
60 REM *                              *
70 REM ********************************
80 TEXT : HOME
90 POKE 103,1: POKE 104,64: POKE 16384,0
100 PRINT  CHR$ (4);"RUN DEFILE.DEMO.PRO
    G"
```

Programme DEFILE.DEMO.PROG

```
10 REM ********************************
20 REM *                              *
30 REM *   Animation graphique - Démo *
40 REM *                              *
50 REM *Patrick QUETTIER - 76190 Yvetot*
60 REM *                              *
70 REM ********************************
80 GOSUB 230: REM  Initialisation
90 GOSUB 160: ON CH% GOSUB 300,350,400,1
   00: GOTO 90: REM  Menu
100 POKE 216,0: PRINT : POP : PRINT D$"P
    R#6": REM  Fin - CH%=4
110 REM ******************************
120 REM Attente clavier
130 VTAB 24: PRINT " Appuyer sur une tou
    che pour continuer";: VTAB 2: CALL B
    E%: POKE KK%,0: WAIT KE%,128: HOME :
     RETURN
140 REM ******************************
150 REM Menu (CH%=1 à 4)
160 HOME : HTAB 9: INVERSE : PRINT  LEFT
    $ (T$,23): HTAB 9: PRINT "*" SPC( 21
    )"*": HTAB 9: PRINT "* ANIMATION GRA
    PHIQUE *"
170 HTAB 9: PRINT "*" SPC( 21)"*": HTAB
    9: PRINT  LEFT$ (T$,23): NORMAL
180 VTAB 10: PRINT T$: PRINT " 1 - Démon
    stration": PRINT : PRINT " 2 - Démon
    stration (bis)": PRINT : PRINT " 3 -
     Mode d'emploi"
190 PRINT : PRINT " 4 - Fin": PRINT : PR
    INT T$: VTAB 24: PRINT  SPC( 8)"Appu
    yer sur 1, 2, 3 ou 4";: CALL BE%: PO
    KE KK%,0
200 WAIT KE%,128: POKE KK%,0:CH% =  PEEK
     (KE%) - 48: ON (CH% < 1 OR CH% > 4)
     GOTO 200: HOME : RETURN
210 REM ******************************
220 REM Initialisation
```

```
230 ONERR  GOTO 610
240 D$ =  CHR$ (4):BE% =  - 1052:KK% =
    - 16368:KE% =  - 16384:T$ = "*******
    ********************************":G
    $ =  CHR$ (34)
250 PRINT D$"BLOAD PARAMETRES+DEFILE.OBJ
    0": PRINT D$"BLOAD DEFILE.DEMO.PIC,A
    8192": PRINT D$"BLOAD SHIFT+ONERR"
260 IF  PEEK (64435) = 234 THEN  CALL 76
    8: REM  On peut supprimer cette  lig
    ne pour un Apple II+ avec ROM LC
270 RETURN
280 REM ********************************
290 REM Démonstration - CH%=1
300 IP$ = "": FOR T = 32 TO 127:IP$ = IP
    $ +  CHR$ (T): NEXT
310 IP$ = "* *  Animation Graphique par
    Patrick QUETTIER - Une ligne de 255
    caractères maxi peut défiler comme a
    ctuellement - Voici tous les caractè
    res disponibles: " + IP$ + "  * *"
320 POKE 49239,0: POKE 49236,0: POKE 492
    34,0: POKE 49232,0: POKE 251,160: CA
    LL 4736: CALL 4736: POKE 49233,0: RE
    TURN
330 REM ********************************
340 REM Démonstration (bis) - CH%=2
350 POKE 49239,0: POKE 49236,0: POKE 492
    35,0: POKE 49232,0: VTAB 21: PRINT "
    Introduisez une phrase quelconque:"
360 VTAB 22: INPUT "";IP$: IF IP$ = "" G
    OTO 360
370 POKE 251,130: CALL 4736: CALL 4736:
    POKE 49233,0: RETURN
380 REM ********************************
390 REM Mode d'emploi - CH%=3
400 PRINT  SPC( 14)"Généralités": VTAB 4
    : PRINT "La routine permettant de fa
    ire de l'ani-mation graphique, est d
    estinée à etre  incorporée à votre
    programme basic. ";
410 PRINT "Pouréviter tout conflit de mé
    moire entre le basic, la page graphi
    que et la routine  en langage machin
    e, votre programme"
420 PRINT "basic devra etre chargé (par
    exemple) audessus de la page graphiq
    ue haute réso- lution 1:"
430 PRINT : PRINT "Au clavier, il faut f
    aire:": PRINT "  POKE 103,1:POKE 104
    ,64:POKE 16384,0      RUN Programme b
    asic"
440 PRINT : PRINT "Ou il faut écrire un
    petit programme    basic qui lancera
     le programme basic    principal:":
    PRINT "  10 POKE 103,1:POKE 104,64:P
    OKE 16384,0  20 PRINT CHR$(4);"G$"RU
    N Programme basic"G$
450 GOSUB 130: PRINT " Initialisation pa
    r le programme basic": VTAB 7: PRINT
     "Votre programme basic devra compor
    ter,  dès le début, une ligne:"
460 PRINT : PRINT : PRINT "100 PRINT CHR
    $(4);"G$"BLOAD PARAMETRES+DEFI    LE
    .OBJ0"G$: PRINT
470 PRINT : PRINT "Le fichier PARAMETRES
    +DEFILE.OBJ0 con-  tient l'intégrali
    té de la routine gra-  phique (=lang
    age machine)": GOSUB 130
480 PRINT "   Utilisation par le program
    me basic": VTAB 5: PRINT "Commencer
    par charger une page graphiquehaute
    résolution (déjà existante) à    pa
    rtir d'une disquette:"
490 PRINT : PRINT "200 PRINT CHR$(4);"G$
    "BLOAD Page graphique,    A8192"G$
500 PRINT : PRINT "On peut aussi réalise
    r un dessin avec  les instructions
    graphiques du basic    (HPLOT etc...
    )"
510 PRINT : PRINT : PRINT "Régler la tab
    ulation verticale de la    ligne à f
    aire défiler:": PRINT : PRINT "210 P
    OKE 251,0 (haut écran) à 176 (bas
       écran)": GOSUB 130
520 VTAB 3: PRINT "Mettre dans la variab
    le alphanumérique  IP$ la phrase que
     l'on veut faire       défiler:": PR
    INT : PRINT "   220 IP$="G$"Bonjour"
    G$: PRINT "ou 220 INPUT IP$"
530 PRINT : PRINT : PRINT "Commuter la p
    age graphique à l'écran:": PRINT : P
    RINT "   230 POKE 49239,0:POKE 49236
    ,0:POKE 49       234,0:POKE 49232,0"
540 PRINT : PRINT : PRINT "Puis lancer l
    e défilement:": PRINT : PRINT "   24
    0 CALL 4736": GOSUB 130
550 PRINT  SPC( 15)"En résumé": VTAB 4:
    PRINT "Voici, comme exemple, un (min
    i) program-me qui utilise la routine
    :"
560 PRINT : PRINT : PRINT "100 PRINT CHR
    $(4);"G$"BLOAD PARAMETRES+DEFI    LE
    .OBJ0"G$: PRINT
570 PRINT "200 PRINT CHR$(4);"G$"BLOAD D
    EFILE.DEMO.PIC    ,A8192"G$: PRINT :
     PRINT "210 POKE 251,160"
580 PRINT : PRINT "220 INPUT IP$": PRINT
     : PRINT "230 POKE 49239,0:POKE 4923
    6,0:POKE 49234    ,0:POKE 49232,0":
    PRINT : PRINT "240 CALL 4736": GOTO
    130
590 REM ********************************
600 REM Traitement de ONERR GOTO
610 CALL 843:E =  PEEK (222): IF E = 255
     THEN  RESUME
620 HOME : VTAB 10: PRINT "Erreur de cod
    e "E" à la ligne " PEEK (218) +  PEE
    K (219) * 256: END
```

Source DEFILE
(Assembleur DOS TOOL-KIT)

```
 1 *
 2 ***********************
 3 *                     *
 4 * ANIMATION GRAPHIQUE *
 5 *                     *
 6 *  Patrick QUETTIER   *
 7 *    76190 Yvetot     *
 8 *                     *
 9 ***********************
10 *
11 *Début:        4736        $12.80
12 *Fin:          4975        $13.6F
13 *Longueur:      240        $00.F0
14 *
15 ******************************************
16 *Variables utilisées par le programme
17 *
18 DB      EQU  4736  *Assemblé à cette adresse
19 HAUT    EQU  251   *1-Tab. vert. haut texte
20 HAUT16  EQU  253   *1-HAUT16=HAUT+16
21 HGR     EQU  254   *2-Adrler octet ligne HGR
22 OCTET1  EQU  252   *1-Partie haute col carac
23 OCTET2  EQU  250   *1-Partie basse col carac
24 CARAC   EQU  249   *1-Compteur 12 col carac
25 ASCII   EQU  235   *2-Adr définition carac
26 ASCII12 EQU  237   *2-ASCII12=ASCII+12
27 LOMEM   EQU  105   *2-Début zone données num
28 LONG    EQU  239   *1-Longu de la chaine IP$
29 POINTE  EQU  215   *1-Compteur carac IP$
30 ADRIP   EQU  206   *2-Adressage ind pour IP$
31 ADRESSE EQU  DB-2688   *Bloc de paramètres
32 *
33 ******************************************
34         ORG  DB        *Adresse d'assemblage
35 *
36 ******************************************
37 *Arrivée du Basic après réglage HAUT et IP$
38 *
39         LDA  HAUT      *HAUT16=HAUT+16
40         CMP  #177
41         BCC  TOTO
42         JMP  $E199     *Illegal quantity er
43 TOTO    ADC  #16
44         STA  HAUT16
45 *
46         JSR  CHERCHE
47         LDY  #2        *LONG=longueur IP$
48         LDA  (ADRIP),Y
49         STA  LONG
50         BNE  LBL13
51         RTS             *FIN retour basic
52 *
53 LBL13   INY
54         LDA  (ADRIP),Y
55         TAX
56         INY
57         LDA  (ADRIP),Y
58         STA  ADRIP+1
59         STX  ADRIP
60 *
61         LDA  #0         *POINTE=0
62         STA  POINTE
63 *
64 LBL14   LDY  POINTE     *Rentre les LONG car
65         LDA  (ADRIP),Y *de IP$ de POINTE=0
66         AND  #127       *à LONG-1
67         JSR  RENTRE
68         INC  POINTE
69         LDA  POINTE
70         CMP  LONG
71         BNE  LBL14
72 *
73         LDA  #24        *POINTE=24 espaces
74         STA  POINTE     *pour finir la ligne
75 LBL15   LDA  #32
76         JSR  RENTRE
77         DEC  POINTE
78         BNE  LBL15
79         RTS             *FIN retour basic
80 *
81 ******************************************
82 *Rentre le car de code ASCII A, décale de 12
83 *colonnes vers la gauche
84 *
85 RENTRE  CMP  #32        *N'affiche pas un car
86         BCC  LBL16      *de controle (<32)
87         SEC             *ASCII=A-32
88         SBC  #32
89         STA  ASCII
90 *
91         LDA  #0         *ASCII=24*ASCII
92         LDX  #8
93         LSR  ASCII
94 LBL8    BCC  LBL9
95         CLC
96         ADC  #24
97 LBL9    ROR  A
98         ROR  ASCII
99         DEX
100        BNE  LBL8
101        STA  ASCII+1
102 *
103        LDA  ASCII      *ASC=ASC+ADRESSE+384
104        CLC             *adr 1er octet de la
105        ADC  #>ADRESSE+384
106        STA  ASCII      *"matrice" du car
107        LDA  ASCII+1
108        ADC  #<ADRESSE+384
109        STA  ASCII+1
110 *
111        STA  ASCII12+1 *ASCII12=ASCII+12
112        LDA  ASCII
113        CLC
114        ADC  #12
115        STA  ASCII12
116        BCC  LBL4
117        INC  ASCII12+1
118 *
119 LBL4   LDA  #0         *CARAC=0 (comp des 12
120        STA  CARAC      *colonnes du car)
121 LBL7   LDY  CARAC      *octet haut colonne
122        LDA  (ASCII),Y
```

```
123          STA  OCTET1
124          LDA  (ASCII12),Y
125          STA  OCTET2     *et l'octet bas
126          JSR  DROIGAU    *Défill col à gauche
127          INC  CARAC      *CARAC=CARAC+1
128          LDA  CARAC      *continue pour 12 col
129          CMP  #12        *du caractère
130          BNE  LBL7
131 LBL16    RTS             *Fin
132 *
133 ****************************************
134 *Cherche la variable IP$
135 *
136 CHERCHE LDA  LOMEM       *On commence à LOMEM
137          STA  ADRIP
138          LDA  LOMEM+1
139          STA  ADRIP+1
140 *
141 LBL10    LDY  #0
142          LDA  #73        *1er octet:"I" ?
143          CMP  (ADRIP),Y
144          BNE  LBL11
145          INY
146          LDA  #208       *Oui-2ème octet:"P" ?
147          CMP  (ADRIP),Y
148          BEQ  LBL12
149 *
150 LBL11    CLC             *Non, ajoute 7
151          LDA  ADRIP
152          ADC  #7
153          STA  ADRIP
154          BCC  LBL10
155          INC  ADRIP+1
156          JMP  LBL10
157 *
158 LBL12    RTS             *Résultat dans ADRIP
159 *
160 ****************************************
161 *Fait défiler 16 li (HAUT à HAUT16-1), d'une
162 *colonne vers la gauche
163 *
164 DROIGAU LDX  HAUT        *X=tab verticale=HAUT
165          CLV             *Drapeau V=0
166 *
167 LBL0     LDA  ADRESSE,X *HGR=adr oct 1 en 0,X
168          STA  HGR+1
169          LDA  ADRESSE+192,X
170          STA  HGR
171 *
172          ROR  OCTET2     *car à entrer en
173          ROR  OCTET1
174          LDY  #39        *Y=tab horiz=39 à 0
175 *
176 LBL1     LDA  (HGR),Y    *A=contenu oct en Y,X
177          BCC  LBL2       *Met retenue en MSB A
178          ORA  #128       *Cas 1
179          BVC  LBL3       *Toujours pris
180 LBL2     AND  #127       *Cas 0
181 LBL3     LSR  A          *Décalage droite
182          STA  (HGR),Y    *A dans octet (Y,X)
183          DEY             *Y=Y-1
184          BPL  LBL1       *meme ligne Y=39 à 0
185          INX             *X=X+1
186          CPX  HAUT16     *pour lignes X=HAUT
187          BNE  LBL0       *à HAUT16-1
188 *
189          LDY  #15        *Temporisation
190 LBL6     LDX  #$FF
191 LBL5     DEX
192          BNE  LBL5
193          DEY
194          BNE  LBL6
195          RTS             *Fin
196 *
197 ****************************************
```

Récapitulation PARAMETRES

```
0800- 20 24 28 2C 30 34 38 3C
0808- 20 24 28 2C 30 34 38 3C
0810- 21 25 29 2D 31 35 39 3D
0818- 21 25 29 2D 31 35 39 3D
0820- 22 26 2A 2E 32 36 3A 3E
0828- 22 26 2A 2E 32 36 3A 3E
0830- 23 27 2B 2F 33 37 3B 3F
0838- 23 27 2B 2F 33 37 3B 3F
0840- 20 24 28 2C 30 34 38 3C
0848- 20 24 28 2C 30 34 38 3C
0850- 21 25 29 2D 31 35 39 3D
0858- 21 25 29 2D 31 35 39 3D
0860- 22 26 2A 2E 32 36 3A 3E
0868- 22 26 2A 2E 32 36 3A 3E
0870- 23 27 2B 2F 33 37 3B 3F
0878- 23 27 2B 2F 33 37 3B 3F
0880- 20 24 28 2C 30 34 38 3C
0888- 20 24 28 2C 30 34 38 3C
0890- 21 25 29 2D 31 35 39 3D
0898- 21 25 29 2D 31 35 39 3D
08A0- 22 26 2A 2E 32 36 3A 3E
08A8- 22 26 2A 2E 32 36 3A 3E
08B0- 23 27 2B 2F 33 37 3B 3F
08B8- 23 27 2B 2F 33 37 3B 3F
08C0- 00 00 00 00 00 00 00 00
08C8- 80 80 80 80 80 80 80 80
08D0- 00 00 00 00 00 00 00 00
08D8- 80 80 80 80 80 80 80 80
08E0- 00 00 00 00 00 00 00 00
08E8- 80 80 80 80 80 80 80 80
08F0- 00 00 00 00 00 00 00 00
08F8- 80 80 80 80 80 80 80 80
0900- 28 28 28 28 28 28 28 28
0908- A8 A8 A8 A8 A8 A8 A8 A8
0910- 28 28 28 28 28 28 28 28
0918- A8 A8 A8 A8 A8 A8 A8 A8
0920- 28 28 28 28 28 28 28 28
0928- A8 A8 A8 A8 A8 A8 A8 A8
0930- 28 28 28 28 28 28 28 28
0938- A8 A8 A8 A8 A8 A8 A8 A8
0940- 50 50 50 50 50 50 50 50
0948- D0 D0 D0 D0 D0 D0 D0 D0
0950- 50 50 50 50 50 50 50 50
0958- D0 D0 D0 D0 D0 D0 D0 D0
0960- 50 50 50 50 50 50 50 50
0968- D0 D0 D0 D0 D0 D0 D0 D0
0970- 50 50 50 50 50 50 50 50
0978- D0 D0 D0 D0 D0 D0 D0 D0
0980- 00 00 00 00 00 00 00 00
0988- 00 00 00 00 00 00 00 00
0990- 00 00 00 00 00 00 00 00
0998- 00 00 00 00 00 FF FF 00
09A0- 00 00 00 00 00 00 00 00
09A8- 00 33 33 00 00 00 00 00
09B0- 00 00 00 0F 0F 00 00 0F
09B8- 0F 00 00 00 00 00 00 00
09C0- 00 00 00 00 00 00 00 00
09C8- 00 C0 C0 FC FE C7 C3 C3
09D0- 07 0E 0C 00 00 0C 18 1F
09D8- 3F 30 30 30 38 1C 0C 00
09E0- 00 00 78 FC CC FF FF CC
09E8- CC 8C 00 00 00 00 0C 0C
09F0- 0C 3F 3F 0C 0F 07 00 00
09F8- 00 0E 1F 1B 1F CE F0 3C
0A00- 0F 03 00 00 00 00 30 3C
0A08- 0F 03 1C 3E 36 3E 1C 00
0A10- 00 00 9E FF E3 F3 3F 1E
0A18- 00 00 00 00 00 1F 3F 31
0A20- 30 31 1B 0E 1C 3E 33 00
0A28- 00 00 00 00 26 3F 1F 0E
0A30- 00 00 00 00 00 00 00 00
0A38- 00 00 00 00 00 00 00 00
0A40- 00 00 00 00 FC FE 03 01
0A48- 00 00 00 00 00 00 00 00
0A50- 0F 1F 30 20 00 00 00 00
0A58- 00 00 00 00 01 03 FE FC
0A60- 00 00 00 00 00 00 00 00
0A68- 20 30 1F 0F 00 00 00 00
0A70- 00 10 30 60 C0 F8 F8 C0
0A78- 60 30 10 00 00 11 19 0D
0A80- 07 3F 3F 07 0D 19 11 00
0A88- 00 00 00 00 00 F0 F0 00
0A90- 00 00 00 00 00 03 03 03
0A98- 03 3F 3F 03 03 03 03 00
0AA0- 00 00 00 00 00 00 00 00
0AA8- 00 00 00 00 00 00 00 00
0AB0- 98 FC 7C 38 00 00 00 00
0AB8- 00 00 00 00 00 00 00 00
0AC0- 00 00 00 08 00 00 00 03
0AC8- 03 03 03 03 00 00 00 00
0AD0- 00 00 00 00 00 00 00 00
0AD8- 00 00 00 00 00 00 00 00
0AE0- 38 38 38 00 00 00 00 00
0AE8- 00 00 00 00 00 C0 F0 3C
0AF0- 0F 03 00 00 00 00 30 3C
0AF8- 0F 03 00 00 00 00 00 00
0B00- 00 FC FE 07 03 C3 F3 3F
0B08- 0F FE FC 00 00 0F 1F 3C
0B10- 3F 33 30 30 38 1F 0F 00
0B18- 00 00 00 06 06 FF FF 00
0B20- 00 00 00 00 00 00 00 30
0B28- 30 3F 3F 30 30 00 00 00
0B30- 00 1C 1E 07 03 03 83 C3
0B38- E7 7E 3C 00 00 30 38 3C
0B40- 3E 37 33 31 30 30 30 00
0B48- 00 0C 0E 07 C3 C3 C3 C3
0B50- E7 FE 3C 00 00 0C 1C 38
0B58- 30 30 30 30 39 1F 0F 00
0B60- 00 C0 E0 70 38 1C 0E FF
0B68- FF 00 00 00 00 03 03 03
0B70- 03 03 03 3F 3F 03 03 00
0B78- 00 7F 7F 63 63 63 63 63
0B80- E3 C3 83 00 00 0C 1C 38
0B88- 30 30 30 30 38 1F 0F 00
0B90- 00 FC FE 87 C3 C3 C3 C3
0B98- C7 8E 0C 00 00 0F 1F 39
0BA0- 30 30 30 30 39 1F 0F 00
0BA8- 00 03 03 03 83 C3 E3 73
0BB0- 3B 1F 0F 00 00 00 00 3F
0BB8- 3F 01 00 00 00 00 00 00
0BC0- 00 3C FE E7 C3 C3 C3 C3
0BC8- E7 FE 3C 00 00 0F 1F 39
0BD0- 30 30 30 30 39 1F 0F 00
0BD8- 00 3C 7E E7 C3 C3 C3 C3
0BE0- 67 FE FC 00 00 0C 1C 38
0BE8- 30 30 30 30 38 1F 0F 00
0BF0- 00 00 00 00 E0 E0 E0 00
```

```
0BF8- 00 00 00 00 00 00 00 00
0C00- 38 38 38 00 00 00 00 00
0C08- 00 00 00 00 E0 E0 E0 00
0C10- 00 00 00 00 00 00 00 00
0C18- B8 F8 78 00 00 00 00 00
0C20- 00 00 00 00 80 C0 60 30
0C28- 10 00 00 00 00 00 00 03
0C30- 07 0C 18 30 20 00 00 00
0C38- 00 00 C0 C0 C0 C0 C0 C0
0C40- C0 C0 00 00 00 00 0C 0C
0C48- 0C 0C 0C 0C 0C 0C 00 00
0C50- 00 00 00 10 30 60 C0 80
0C58- 00 00 00 00 00 00 00 20
0C60- 30 18 0C 07 03 00 00 00
0C68- 00 0C 0E 07 03 83 C3 C3
0C70- E7 7E 3C 00 00 00 00 00
0C78- 00 33 33 00 00 00 00 00
0C80- 00 C0 E0 70 31 33 36 34
0C88- F0 F0 00 00 00 0F 1F 38
0C90- 30 30 30 18 3F 3F 00 00
0C98- 00 F0 F8 1C 0E 07 07 0E
0CA0- 1C F8 F0 00 00 3F 3F 03
0CA8- 03 03 03 03 03 3F 3F 00
0CB0- 00 FF FF C3 C3 C3 C3 C3
0CB8- E7 BE 1C 00 00 3F 3F 30
0CC0- 30 30 30 30 39 1F 0F 00
0CC8- 00 FC FE 07 03 03 03 03
0CD0- 07 0E 0C 00 00 0F 1F 38
0CD8- 30 30 30 30 38 1C 0C 00
0CE0- 00 FF FF 03 03 03 03 03
0CE8- 07 FE FC 00 00 3F 3F 30
0CF0- 30 30 30 30 38 1F 0F 00
0CF8- 00 FF FF C3 C3 C3 C3 C3
0D00- C3 C3 03 00 00 3F 3F 30
0D08- 30 30 30 30 30 30 30 00
0D10- 00 FF FF C3 C3 C3 C3 C3
0D18- C3 C3 03 00 00 3F 3F 00
0D20- 00 00 00 00 00 00 00 00
0D28- 00 FC FE 07 03 03 83 83
0D30- 87 8E 8C 00 00 0F 1F 38
0D38- 30 30 31 31 31 3F 3F 00
0D40- 00 FF FF C0 C0 C0 C0 C0
0D48- C0 FF FF 00 00 3F 3F 00
0D50- 00 00 00 00 00 3F 3F 00
0D58- 00 00 00 00 03 FF FF 03
0D60- 00 00 00 00 00 00 00 00
0D68- 30 3F 3F 30 00 00 00 00
0D70- 00 00 00 03 03 03 03 FF
0D78- FF 03 03 00 00 0E 1E 38
0D80- 30 30 38 1F 0F 00 00 00
0D88- 00 FF FF E0 F0 B8 1C 0E
0D90- 07 03 00 00 00 3F 3F 00
0D98- 01 03 07 0E 1C 38 30 00
0DA0- 00 00 FF FF 00 00 00 00
0DA8- 00 00 00 00 00 00 3F 3F
0DB0- 30 30 30 30 30 30 00 00
0DB8- 00 FF FF 3C F0 C0 C0 F0
0DC0- 3C FF FF 00 00 3F 3F 00
0DC8- 00 01 01 00 00 3F 3F 00
0DD0- 00 FF FF 0F 3C F0 C0 00
0DD8- 00 FF FF 00 00 3F 3F 00
0DE0- 00 00 03 0F 3C 3F 3F 00
0DE8- 00 FC FE 07 03 03 03 03
0DF0- 07 FE FC 00 00 0F 1F 38
0DF8- 30 30 30 30 38 1F 0F 00
0E00- 00 FF FF 83 83 83 83 83
0E08- C7 FE 7C 00 00 3F 3F 01
0E10- 01 01 01 01 01 00 00 00
0E18- 00 FC FE 07 03 03 03 03
0E20- 07 FE FC 00 00 0F 1F 38
0E28- 30 32 36 3E 1C 3F 2F 00
0E30- 00 FF FF 83 83 83 83 83
0E38- C7 FE 7C 00 00 3F 3F 01
0E40- 01 03 07 0F 1D 38 30 00
0E48- 00 3C 7E E7 C3 C3 C3 C3
0E50- C7 8E 0C 00 00 0C 1C 38
0E58- 30 30 30 30 39 1F 0F 00
0E60- 00 03 03 03 03 FF FF 03
0E68- 03 03 03 00 00 00 00 00
0E70- 00 3F 3F 00 00 00 00 00
0E78- 00 FF FF 00 00 00 00 00
0E80- 00 FF FF 00 00 0F 1F 38
0E88- 30 30 30 30 38 1F 0F 00
0E90- 00 3F FF C0 00 00 00 00
0E98- C0 FF 3F 00 00 00 00 03
0EA0- 0F 3C 3C 0F 03 00 00 00
0EA8- 00 FF FF 00 00 00 00 00
0EB0- 00 FF FF 00 00 03 0F 3C
0EB8- 30 1E 1E 30 3C 0F 03 00
0EC0- 00 0F 1F 30 E0 C0 C0 E0
0EC8- 30 1F 0F 00 00 3C 3E 03
0ED0- 01 00 00 01 03 3E 3C 00
0ED8- 00 1F 3F 70 E0 C0 C0 E0
0EE0- 70 3F 1F 00 00 00 00 00
0EE8- 00 3F 3F 00 00 00 00 00
0EF0- 00 03 03 03 83 C3 E3 73
0EF8- 3B 1F 0F 00 00 3C 3E 3F
0F00- 33 31 30 30 30 30 30 00
0F08- 00 00 00 78 FC 84 84 FC
0F10- 78 00 00 00 00 00 00 00
0F18- 00 00 00 00 00 00 00 00
0F20- 00 C0 E0 70 30 30 30 30
0F28- 70 60 40 00 00 0F 1F 38
0F30- F0 F0 30 30 38 18 08 00
0F38- 00 00 00 DC FE 33 21 E1
0F40- C7 06 00 00 00 00 18 38
0F48- 21 21 33 1F 0E 00 00 00
0F50- 00 00 20 30 18 0C 0C 18
0F58- 30 20 00 00 00 00 00 00
0F60- 00 00 00 00 00 00 00 00
0F68- 00 00 00 00 00 00 00 00
0F70- 00 00 00 00 18 18 18 18
0F78- 18 18 18 18 18 18 18 18
0F80- 00 00 00 00 0E 1F 3F 26
0F88- 00 00 00 00 00 00 00 00
0F90- 00 00 00 00 00 00 00 00
0F98- 00 C0 E0 70 30 30 30 30
0FA0- F0 F0 00 00 00 0F 1F 38
0FA8- 30 30 30 18 3F 3F 00 00
0FB0- 00 FF FF 60 30 30 30 30
0FB8- 70 E0 C0 00 00 3F 3F 18
0FC0- 30 30 30 30 38 1F 0F 00
0FC8- 00 C0 E0 70 30 30 30 30
0FD0- 70 60 40 00 00 0F 1F 38
0FD8- 30 30 30 30 38 18 08 00
0FE0- 00 C0 E0 70 30 30 30 30
0FE8- 30 FF FF 00 00 0F 1F 38
0FF0- 30 30 30 30 18 3F 3F 00
0FF8- 00 C0 E0 70 30 30 30 30
1000- 70 E0 C0 00 00 0F 1F 3B
1008- 33 33 33 33 33 33 33 00
1010- 00 00 60 60 FC FE 67 63
1018- 63 03 00 00 00 00 00 00
1020- 3F 3F 00 00 00 00 00 00
1028- 00 C0 E0 70 30 30 30 30
1030- 70 F0 F0 00 00 03 C7 CE
1038- CC CC CC CC E6 7F 3F 00
1040- 00 FF FF 60 30 30 30 70
1048- E0 C0 00 00 00 3F 3F 00
1050- 00 00 00 00 3F 3F 00 00
1058- 00 00 00 00 00 F3 F3 00
1060- 00 00 00 00 00 00 00 00
1068- 00 3F 3F 00 00 00 00 00
1070- 00 00 00 00 30 30 F3 F3
1078- 00 00 00 00 00 00 00 00
1080- C0 E0 7F 3F 00 00 00 00
1088- 00 00 FF FF 00 80 C0 60
1090- 30 10 00 00 00 00 3F 3F
1098- 03 07 0C 18 30 20 00 00
10A0- 00 00 00 00 00 FF FF 00
10A8- 00 00 00 00 00 00 00 00
10B0- 00 3F 3F 00 00 00 00 00
10B8- 00 F0 F0 20 30 F0 E0 30
10C0- 30 F0 E0 00 00 3F 3F 00
10C8- 00 3F 3F 00 00 3F 3F 00
10D0- 00 F0 F0 60 30 30 30 70
10D8- E0 C0 00 00 00 3F 3F 00
10E0- 00 00 00 00 3F 3F 00 00
10E8- 00 C0 E0 70 30 30 30 70
10F0- E0 C0 00 00 00 0F 1F 38
10F8- 30 30 30 38 1F 0F 00 00
1100- 00 F0 F0 60 30 30 30 70
1108- E0 C0 00 00 00 FF FF 0C
1110- 18 18 18 1C 0F 07 00 00
1118- 00 C0 E0 70 30 30 30 60
1120- F0 F0 00 00 00 07 0F 1C
1128- 18 18 18 0C FF FF 00 00
1130- 00 00 F0 F0 60 30 30 70
1138- 60 00 00 00 00 00 3F 3F
1140- 00 00 00 00 00 00 00 00
1148- 00 00 E0 F0 30 30 30 30
1150- 30 30 00 00 00 00 31 33
1158- 33 33 33 33 3F 1E 00 00
1160- 00 00 30 30 FE FE 30 30
1168- 30 00 00 00 00 00 00 00
1170- 1F 3F 30 30 30 00 00 00
1178- 00 F0 F0 00 00 00 00 00
1180- F0 F0 00 00 00 0F 1F 38
1188- 30 30 30 18 3F 3F 00 00
1190- 00 30 F0 C0 00 00 00 00
1198- C0 F0 30 00 00 00 00 03
11A0- 0F 3C 3C 0F 03 00 00 00
11A8- 00 F0 F0 00 00 00 00 00
11B0- 00 F0 F0 00 00 0F 1F 38
11B8- 18 0E 0E 18 38 1F 0F 00
11C0- 00 30 70 E0 C0 80 80 C0
11C8- E0 70 30 00 00 30 38 1C
11D0- 0F 07 07 0F 1C 38 30 00
11D8- 00 F0 F0 00 00 00 00 00
11E0- F0 F0 00 00 00 C3 C7 CE
11E8- CC CC CC E6 7F 3F 00 00
11F0- 00 30 30 30 30 30 B0 F0
11F8- 70 30 00 00 00 30 38 3C
1200- 36 33 31 30 30 30 00 00
1208- 00 C0 E0 70 34 36 33 31
1210- 70 E0 C0 00 00 0F 1F 3B
1218- 33 33 33 33 33 33 33 00
1220- 00 F0 F0 01 03 06 04 00
1228- F0 F0 00 00 00 0F 1F 38
1230- 30 30 30 18 3F 3F 00 00
1238- 00 C0 E0 70 31 33 36 34
1240- 70 E0 C0 00 00 0F 1F 3B
1248- 33 33 33 33 33 33 33 00
1250- 00 00 00 06 06 00 00 06
1258- 06 00 00 00 00 00 00 00
1260- 00 00 00 00 00 00 00 00
1268- 00 A8 50 A8 50 A8 50 A8
1270- 50 A8 00 00 00 2A 15 2A
1278- 15 2A 15 2A 15 2A 00 00
```

Récapitulation DEFILE.OBJ0

```
1280- A5 FB C9 B1 90 03 4C 99
1288- E1 69 10 85 FD 20 17 13
1290- A0 02 B1 CE 85 EF D0 01
1298- 60 C8 B1 CE AA C8 B1 CE
12A0- 85 CF 86 CE A9 00 85 D7
12A8- A4 D7 B1 CE 29 7F 20 C7
12B0- 12 E6 D7 A5 D7 C5 EF D0
12B8- EF A9 18 85 D7 A9 20 20
12C0- C7 12 C6 D7 D0 F7 60 C9
12C8- 20 90 4B 38 E9 20 85 EB
12D0- A9 00 A2 08 46 EB 90 03
12D8- 18 69 18 6A 66 EB CA D0
12E0- F5 85 EC A5 EB 18 69 80
12E8- 85 EB A5 EC 69 09 85 EC
12F0- 85 EE A5 EB 18 69 0C 85
12F8- ED 90 02 E6 EE A9 00 85
1300- F9 A4 F9 B1 EB 85 FC B1
1308- ED 85 FA 20 3D 13 E6 F9
1310- A5 F9 C9 0C D0 EB 60 A5
1318- 69 85 CE A5 6A 85 CF A0
1320- 00 A9 49 D1 CE D0 07 C8
1328- A9 D0 D1 CE F0 0E 18 A5
1330- CE 69 07 85 CE 90 E8 E6
1338- CF 4C 1F 13 60 A6 FB B8
1340- BD 00 08 85 FF BD C0 08
1348- 85 FE 66 FA 66 FC A0 27
1350- B1 FE 90 04 09 80 50 02
1358- 29 7F 4A 91 FE 88 10 F0
1360- E8 E4 FD D0 DB A0 0F A2
1368- FF CA D0 FD 88 D0 F8 60
```

Récapitulation SHIFT+ONERR

```
0300- A9 0F 85 36 A9 03 85 37
0308- A9 00 85 48 4C EA 03 8D
0310- 4A 03 08 8A 48 AD 4A 03
0318- A2 07 DD 3A 03 F0 0C CA
0320- 10 F8 C9 E0 90 08 38 E9
0328- 20 D0 03 BD 42 03 8D 4A
0330- 03 68 AA 28 AD 4A 03 4C
0338- F0 FD C0 DC E0 FB FC FD
0340- FE FF C1 C3 A1 C5 D5 C5
0348- AD A1 00 68 A8 68 A6 DF
0350- 9A 48 98 48 60
```

Accompagné d'une cinquantaine de pages de documentation, δisk Manager permet de recréer les commandes du Dos, redéfinir l'organisation d'une disquette, grâce à un jeu d'instructions qui en fait un langage simple d'accès à la disquette. Il offre également un programme simple d'édition à l'aide de commandes évoluées. 4 utilitaires figurent aussi sur la disquette :

Utili-disque : reconstruction d'une disquette détruite, Vérification, Plan d'occupation
Ultra-copie : pour un backup particulièrement rapide
Edicat : Edition du catalogue, classement des fichiers, Titres...
Multi-disque : pour le classement de tous vos programmes (tri instantané).

δisk Manager, le Dos en Kit

de Dan Steevey

Calcul de π

Michel Dominique

Pour tous ceux qui estiment péremptoire l'affirmation d'Olivier Herz dans son article "EXP. (π*SQR (163)) est-il entier ?" (Pom's 12), selon laquelle π est supposé connu, voici un petit programme en Applesoft, écrit pour calculer ce nombre.

Pour ce faire, nous utiliserons le résultat mathématique suivant :

$\pi/2=1+(1/3)+(1*2)/(3*5)+...+(n!/1*3...*(2n+1))+...$

Cette méthode compte parmi les plus simples, mais elle permet un calcul rapide par ordinateur. L'algorithme utilisant la méthode de Horner consiste à calculer par approximations successives et en s'arrêtant au rang n :

$\pi=2+(1/3)*(2+(2/5))*(2+(3/7))*(2+(4/9))*(2+...*(2+(n/2n+1))$

Le programme commence par déterminer le nombre de termes calculés (lignes 30 et 70); l'expérience montre qu'il s'agit environ des 10/3 du nombre de décimales voulu. Puis il effectue le calcul dans la courte boucle de lignes 100 à 250, ce qui prend, en secondes, un temps équivalent au 25ème du carré du nombre de décimales. Le reste du programme est consacré à l'affichage et au stockage des décimales sur disque (fichier TEXT).

Remarques

1) Ce calcul se programme en Basic (8 décimales) par :

```
PI=0:FOR N=30 TO 1
STEP-1:PI=PI*N/(2*
N+1)+2:NEXT:PRINT
PI
```

2) On peut aussi effectuer le calcul avec le Calculateur Entier de Jacques Duma (Pom's 12), par exemple avec 100 décimales :

```
K = 10^100: K=2*K:
N=333: I=2*N: I=I+
1 : P=0 111#: 3#:
P=P*N: P=P/I: P=P+
K: N=N-1: I=I-2 ?P
```

Notons que ce calcul prend près de 50 mn, contre 6 mn 30 au programme Basic avec le même nombre de décimales. En effet, le Calculateur Entier n'est pas plus rapide pour le produit d'un grand nombre par un petit que pour le produit de deux grands nombres, contrairement à un programme Basic que l'on peut adapter à la longueur des nombres.

3) Le programme ne fonctionne correctement que pour un calcul avec au moins six décimales.

NDLR : Pour aller plus avant dans la connaissance de π, le supplément au 'Petit Archimède' n° 64-65 de mai 1980 consacre 280 pages à une cinquantaine de méthodes de calcul et en prime les 25000 premières décimales pour vérifier vos calculs. Cet ouvrage est en vente à la librairie du Palais de la Découverte à Paris.

3,14159265358979323846264338327950288417169399375105820974944592307816406286208998628034825342117067982148086513282306647093844609550582231725359408128481117450 28

Programme PI

```
0 REM ********************
1 REM CALCUL DES DECIMALES
2 REM          DE PI
3 REM    MICHEL DOMINIQUE
4 REM ********************
5 GOSUB 5000
10 TEXT : HOME : CLEAR : IN
   PUT "NOMBRE DE DECIMALES
   : ";NDEC$:NDEC =  VAL (N
   DEC$): IF NDEC$ = "" THE
   N  END
12 ON NDEC < 10 GOTO 10
24 REM ********************
25 REM    CALCUL DU NOMBRE
26 REM        DE BOUCLES
27 REM ********************
30 VTAB 4: HTAB 14: PRINT
   "PATIENCE...":NMEM =  I
   NT (NDEC / 5) + ( INT (
   NDEC / 5 <  > NDEC / 5)
   ):N = (NDEC + 1) *  LOG
    (10):S = 0:NB = 1
70 NB = NB + 1:S = S +  LO
   G (2 + 1 / NB): IF N >
    = S THEN 70

80 DIM DEC(NMEM):DEC(1) =
   100000:NI = DEC(1):NJ =
    NI * 10:AD = 2 * NI
89 REM ********************
90 REM CALCUL DES DECIMALES
91 REM ********************
100 FOR N = NB TO 1 STEP
     - 1: VTAB 10: HTAB 15:
     PRINT NB - N + 1;"/";
    NB:M = 0:K = 2 * N + 1
120 FOR I = 1 TO NMEM:Z =
    DEC(I) + M:DEC(I) =  I
    NT (Z / K):M = (Z - DE
    C(I) * K) * NI: NEXT I
170 M = 0
180 FOR I = NMEM TO 2 STEP
     - 1:DEC(I) = DEC(I)
    * N + M:M =  INT (DEC(
    I) / NI):DEC(I) = DEC(
    I) - M * NI: NEXT I
230 DEC(1) = DEC(1) * N +
    M:M =  INT ( INT (1) /
     NJ):DEC(1) = DEC(1) -
     M * NJ + AD
260 NEXT N
270 GOSUB 1000: GOSUB 4000
```

```
    : GOTO 10
991 REM ******************
993 REM      AFFICHAGE
994 REM    DES DECIMALES
996 REM ******************
1000 HOME : FOR I = 1 TO 3:
     CALL  - 1052: NEXT
1010 PRINT  TAB( 5);"*** PI
     AVEC ";NDEC;" DECIMAL
     ES ***": POKE 34,3: HO
     ME
1020 C$ =  STR$ (DEC(1)):C$
     =  RIGHT$ (C$,5):DEC(
     1) =  VAL (C$)
1030 PRINT "3, ";:N = 0:L =
     0
1040 FOR I = 1 TO NMEM - 1
1050 IF N = 6 THEN N = 0: P
     RINT : PRINT  SPC( 3);
     :L = L + 1
1060 Z$ =  STR$ (DEC(I))
1070 IF  LEN (Z$) < 5 THEN
     Z$ = "0" + Z$: GOTO 10
     70
1080 PRINT Z$;" ";:N = N +
     1
1090 NEXT I
1100 LL = NDEC - 5 * (NMEM
     - 1): IF  NOT LL THEN
     1140
1110 Z$ =  STR$ (DEC(NMEM))
1120 IF  LEN (Z$) < 5 THEN
     Z$ = "0" + Z$: GOTO 11
     20
1130 PRINT  LEFT$ (Z$,LL)
1140 INVERSE : VTAB 23: HTA
     B 3: INPUT "PRESSEZ <R
     ETURN> POUR CONTINUER"
     ;KEY$: NORMAL : RETURN
3991 REM ******************
3993 REM   SAUVEGARDE DES
3994 REM      DECIMALES
3996 REM ******************
4000 PRINT " VOULEZ - VOUS
     UNE SAUVEGARDE ? ";: G
     ET H$
4010 IF H$ <  > "O" AND H$
     <  > "N" THEN  VTAB 4:
      GOTO 4000
4020 IF H$ = "N" THEN  RETU
     RN
4030 HOME :D$ =  CHR$ (4)
4040 PRINT " NOUS ALLONS SA
     UVER LES DECIMALES DE
     PI "
4050 PRINT "CALCULEES SUR D
     ISQUETTE."
4055 PRINT
4060 PRINT "INTRODUISEZ UNE
      DISQUETTE PUIS PRESSE
     Z"
4070 INPUT " 'RETURN '.";H$
4100 PRINT D$"OPEN PIDEC":
     PRINT D$"DELETEPIDEC"
4110 PRINT D$"OPENPIDEC
4120 PRINT D$"WRITEPIDEC
4125 PRINT NDEC: PRINT NMEM
4130 FOR DEC = 1 TO NMEM
4140 PRINT DEC(DEC)
4150 NEXT
4160 PRINT D$"CLOSE
4162 PRINT : INPUT "SAUVEGA
     RDE DE SECURITE ? ";H$
4165 IF H$ <  > "O" AND H$
     <  > "N" THEN 4162
4167 IF H$ = "O" THEN 4055
4170 RETURN
4991 REM ******************
4993 REM   AFFICHAGE DES
4994 REM      DECIMALES
4995 REM   DEJA TROUVEES
4997 REM ******************
5000 TEXT : HOME : PRINT
5010 PRINT "VOULEZ-VOUS VOI
     R LES DECIMALES": PRIN
     T "DEJA TROUVEES ";
5020 INPUT R$: IF R$ <  > "
     O" AND R$ <  > "N" THE
     N 5020
5030 IF R$ = "O" THEN  GOSU
     B 6000
5040 PRINT : PRINT
5050 RETURN
6000 HOME :D$ =  CHR$ (4)
6010 PRINT D$"OPEN PIDEC"
6020 PRINT D$"READ PIDEC"
6030 INPUT NDEC
6040 INPUT NMEM
6045 DIM DEC(NMEM)
6050 FOR DEC = 1 TO NMEM
6060 INPUT DEC(DEC)
6070 NEXT
6080 PRINT D$"CLOSE PIDEC"
6100 GOSUB 1000
6120 RETURN
```

Max : le moniteur étendu

Jacques Supernant

Apple II+, //e, //c

Ce moniteur autorise un contrôle de l'exécution des routines en langage machine

Un mode Trace et Pas à Pas très évolués et sélectifs sont complétés par un accès direct aux registres du 6502 (ou 65C02).

La gestion des fenêtres d'écran simplifie le mode trace.
Une routine permet la recherche de suites d'octets.

Un mini-assembleur très souple fait partie de Max.
Une ligne de commande peut devenir une boucle avec l'ordre JUMP

Les fichiers source sont sur la disquette.

Disquette et documentation : 150,00 F TTC franco . Bon de commande page74.

Bibliographie

Alexandre Duback

Guide Magnard du tout logiciel – 1985/1986 – 415 pages – 149 FF.

Dix programmes par page sont décrits schématiquement dans des cartouches bien présentées, mais contenant une information bien entendu schématique. A ne pas lire si vous cherchez le moindre avis sur les programmes proposés. Le titre est totalement abusif : ce "tout logiciel" parvient à ignorer AppleWorks, AppleWriter, Multiplan, ... De toute façon, avec moins de 200 programmes sur Apple II, on ne pourrait même pas présenter tous les programmes publiés par Pom's pour ses lecteurs depuis le N°1 !

dBase II – Applications de Christophe Stehly, Sybex – 268 pages – 148 FF.

Le livre a été réalisé en utilisant la version 2.4 francisée de dBase II. Une première partie illustre les possibilités du programme en gestion de fichiers, la programmation de dBase II étant couverte dans une seconde partie, à l'aide d'exemples de gestion variés. Un dernier chapitre fort intéressant explique comment il faut s'y prendre pour bien constituer ses fichiers. Ouvrage clair et bien présenté.

La troisième pomme de Jean-Louis Gassée, Hachette – 200 pages – 64 FF.

Le sous-titre, Micro-informatique et Révolution Culturelle, décrit mieux le contenu de cet ouvrage que nous devons à l'ancien directeur d'Apple France, vice-président depuis quelque mois d'Apple en Californie. Lecture indispensable pour comprendre le genre de vocation qu'a pu susciter l'esprit Apple, et dont nous espérons qu'il continuera à en susciter ...

Les écrans du Apple IIe par Phil Robinson, Hachette Informatique – Deux tomes de 26 pages chacun – 110 FF pièce. Traduit de l'anglais.

Il fallait oser un tel titre ... Présentation luxueuse : de la couleur, de la mise en page, des thèmes couverts chacun en une page simple ou double. Bien fait. Ceci dit, en 2 fois 64 pages, on ne peut en aucun cas être exhaustif.

Apple IIc User Guide par Gary Phillips et Donald Scellato, Brady (Prentice Hall) – 370 pages – $20.75.

Ouvrage couvrant tout ce qui peut toucher de près ou de loin les utilisateurs du IIc. Tout d'abord, 146 pages sur le IIc lui-même dans lesquelles on trouve par exemple un chapitre sur l'utilisation du disque System Utilities et un chapitre sur DOS et ProDOS avec une intéressant comparaison instruction par instruction. Ensuite, une cinquantaine de pages de guide pour l'acheteur de logiciel, et enfin un guide d'achat pour les périphériques. Bien réalisé, mais à consulter pour voir si c'est bien ce que vous recherchez.

Destination Aventure Apple IIe, II+ et IIc de Delton Horn, Editions du PSI – 240 pages – 140 FF. Traduction.

La façon dont on programme un jeu d'aventure est expliquée en détail autour d'un premier jeu. C'est clair et détaillé, mais pas forcément performant au niveau de la programmation. Ainsi, à un endroit, on teste la même chaîne sur une cinquantaine de valeurs possibles, au lieu d'effectuer une recherche dichotomique dans une table alpha : cela fait en moyenne 25 comparaisons au lieu de 6 ! Trois autres jeux sont listés en fin d'ouvrage, accompagnés d'explications et d'organigrammes.

Multiplan sur Macintosh de Goulven Habasque, Sybex – 233 pages – 148 FF.

La première partie concerne l'initiation à Multiplan sur Macintosh, avec seulement 53 pages très aérées : ce n'est donc pas pour vous initier à Multiplan que nous vous conseillons cet ouvrage; l'initiation n'est pas assez approfondie à notre avis. Ce qui est intéressant, c'est la suite : vingt-sept petits tableaux d'analyse numérique (6), de géométrie (6), de statistiques (6), d'économie (5) et de gestion (4). Nous vous conseillons de consulter ce livre avant de l'acheter, pour voir si les exemples vous intéressent.

BASIC Engineering, Science, and Business Programs for the Apple II and IIe, Brady Comm. Co, Prentice-Hall – 330 pages – $29.95

Malgré son titre anglais très général, ce livre traite avant tout de divers problèmes de statistiques : gestion des matrices, régressions linéaires, systèmes d'équations multiples, tris, résolution d'équations différentielles, simulations, etc.

Chacun des problèmes est abordé d'abord théoriquement et par l'étude d'exemples, puis par un programme en Applesoft, facilement transposable d'ailleurs sur d'autres ordinateurs. En revanche, aucune routine en assembleur 6502 n'est proposée.

Macintosh, quels logiciels ? de Pierre Courbier, ETSF –143 pages.

Démonstration, de rapide à très schématique, d'un certain nombre de logiciels pour Macintosh. Un fourre-tout bien rédigé, mais dont l'utilité ne nous semble pas évidente. Réalisé sur ImageWriter, mais en qualité courrier ...

MacAstuces par Hervé Thiriez, Editions du PSI – 264 pages – 150 FF.

Hhmm ! Difficile de parler du petit dernier du rédacteur en chef. Le mieux est peut-être de lui demander de le présenter lui-même :
"Ce livre est destiné à tous les utilisateurs du Macintosh ayant déjà fait le tour de leur machine. On y trouve en effet de nombreuses explications sur des notions parfois mal maîtrisées (le rôle du Finder, par exemple), et de multiples trucs et astuces d'utilisation.

Une première partie (88 pages) couvre le Macintosh lui-même : les "allergies" du Mac, le système, le bureau, le clavier, les disquettes et les configurations. La seconde partie – les logiciels – traite de douze logiciels standards. L'avenir est couvert dans la troisième partie, avec les intégrés, accélérateurs et autres programmes faciliteurs.

Une disquette d'accompagnement facultative (200 FF TTC) fournit un ensemble de règles pour MacWrite, un utilitaire pour Multiplan, des démos animées ThinkTank et MacDraw, des images MacPaint bien utiles, un package de programmes statistiques et graphiques et une police de caractères graphiques créée par l'auteur.

Entièrement réalisé en Macintosh, mais sorti sur LaserWriter, ce livre est bien présenté, avec de nombreuses illustrations. Quant au contenu... un must, je l'espère, pour ceux qui veulent tirer le maximum de leur Macintosh !"

Apple IIc, guide de l'utilisateur de Thomas Blackadar, Sybex – 147 pages – 98 FF. Encore une traduction ...

Cet ouvrage est très général et apporte un début d'initation au Basic, donnant à la fin un guide de référence du Basic limité aux seuls mots réservés introduits dans le livre (?). A consulter pour voir si on ne préfère pas lire la documentation Apple et faire des économies.

Multiplan et Chart sur Macintosh de Xavier Bouilloux, Edimicro – 235 pages – 159 FF.

Cela m'énerve toujours de voir un livre sorti sur ImageWriter sans même utiliser la qualité courrier. Certes, cela gagne du temps, mais quand même. La première partie, guide de l'utilisateur pour le livre, Macintosh, Multiplan et Chart, est très succincte : à peine 30 pages pour l'ensemble des deux logiciels ; ce n'est pas comme cela qu'on apprendra à les utiliser. La seconde partie comporte 10 exemples, dont 5 avec passages au graphique. L'intérêt principal de l'ouvrage tient à l'illustration de l'utilisation de Chart, sur lequel il existe moins d'ouvrages. Nous aurions préféré voir un livre uniquement sur Chart, avec un chapitre sur la communication avec Multiplan.

Les ressources de l'Apple IIc de Nicole Bréaud-Pouliquen, Editions du PSI – 106 pages – 85 FF.

Le premier ouvrage en français où l'on trouve un certain nombre de renseignements pour la programmation du IIc, avec en particulier la liste des instructions du 65C02. A conseiller à tous ceux qui programment en assembleur sur le IIc et, aussi, à ceux qui auront mis le chip du IIc sur leur Apple II.

Multiplan facile de Georges Liégois, Marabout – 126 pages – 29 FF.

Un exemple typique de travail rapide : le livre est écrit autour de la version américaine de Multiplan, pour "tous" matériels. C'est ainsi que les tableaux sont sortis en Macintosh, alors que le mode d'emploi se réfère aux commandes de Multiplan avec les autres matériels. Le souci de ratisser large amène l'auteur à présenter les commandes communes à tous les matériels, sauf au Macintosh (pourtant cité en première et quatrième de couverture) pour lequel Multiplan s'utilise très différemment. Avec cette approche "tous matériels", le lecteur est privé de la présentation des raccourcis disponibles sur tel ou tel matériel.

Basic Applesoft Lexicum, (Manuel de référence alphabétique Basic - DOS 3.3 - ProDOS) de Paul Merry, Mnémodyne – Un pavé – 168 FF.

Cet ouvrage date d'un an, mais nous n'avions pas encore eu l'occasion de le voir. Il ajoute à l'ancien manuel de référence, déjà couvert dans Pom's, 62 pages sur le ProDOS, bien utiles dans la mesure où la littérature relative au ProDOS est encore pauvre. Enfin, maintenant, la reliure est un peu plus solide ...

Macintosh : outils, progiciels, applications de Xavier Gaucherand, Edimicro – 237 pages – 148 FF.

Un des rares ouvrages généraux sur le Macintosh qui ne se contente pas de refaire le mode d'emploi de base du Mac avec ses logiciels "résidents" et Multiplan. Cinq parties : présentation du matériel, systèmes et langages, logiciels d'application, domaines d'utilisation et choix d'une configuration. Les analyses de points forts et points faibles des logiciels sont assez bien vues. Un livre intéressant.

Le secrétariat par traitement de texte de Ilya Virgatchik, Marabout – 287 pages – 160 FF.

Nous sommes toujours dans les produits sur CP/M : bien que cela ne soit pas apparent dans le titre, il s'agit d'un manuel pratique, et complet, de Wordstar.

Le manuel du Basic par M. Waite et M. Pardee, Hachette Informatique – 344 p. – 130 FF. Traduit de l'anglais.

Il s'agit d'un manuel de cours, et non d'un livre de référence. Le cours est bien fait, clair et pédagogique, agrémenté par de nombreux petits dessins humoristiques que j'ai bien appréciés. Bonne lecture pour débutant. Il s'agit cependant du Basic Microsoft : n'achetez donc pas ce livre si vous vous intéressez exclusivement à l'Applesoft ou si vous cherchez des informations sur les particularismes des Basics sur Apple.

Courrier des Lecteurs

Alexandre Avrane, Olivier Herz

Possesseur depuis peu d'un Apple IIc, et tout nouvel abonné de votre revue, je viens de découvrir par hasard des signes étranges que je vois parfois utiliser dans certains logiciels. Je me suis par la suite aperçu que je pouvais les faire apparaître en me mettant en mode INVERSE. Comment puis-je utiliser correctement ces signes ?

D'autre part, tous les conseils (POKEs entre autre) donnés auparavant dans votre revue, sont-ils valables sur IIc ? Enfin, pouvez-vous me conseiller un ouvrage qui permette d'approfondir les possibilités de mon Apple préféré ?

Gérald Lamy
84210 La Roque/Pernes

Les caractères souris que vous avez découvert par hasard sur votre Apple IIc sont disponibles directement sur cette machine afin de pouvoir y utiliser la philosophie Macintosh (souris et menus déroulants).

Un exemple de démonstration et de nombreux autres renseignements techniques que vous recherchez sont indiqués dans le livre "Les Ressources de l'Apple IIc" (Editions du PSI).

En ce qui concerne les différents conseils publiés par Pom's depuis le premier numéro, ceux-ci sont utilisables tels quels lorsqu'ils appellent le DOS, ProDOS ou l'Interpréteur Applesoft. En revanche, certaines adresses du moniteur Autostart, de la carte contrôleur de disques, et des interrupteurs entrée/sortie ne sont pas applicables directement sur un IIc.

J'ai créé, avec une table de formes (Shape de l'Applesoft), une carte d'Europe que je charge par un "Bload Europe, A8192". J'utilise ensuite un programme Basic pour repérer les coordonnés des pays.

Ayant, pour l'instant, entré les 2/3 des coordonnées, je m'aperçois que la page graphique se transforme à chaque nouvel ligne de programme : des traits horizontaux apparaissent dans tous les sens. J'ai essayé de mettre un LOMEM:16384, mais sans résultat. Quelle est l'erreur ?

Laurent Poppeschi
76200 Dieppe

Celle-ci provient très certainement de la taille de votre programme Basic. En effet, la localisation en mémoire vive de celui-ci déborde sur les adresses des pages graphiques haute résolution.

Une première solution consiste à charger votre image graphique en 16384 ($4000) et non 8192 ($2000) et à utiliser HGR2 au lieu de HGR.

Deuxième possibilité : réduisez la taille de votre programme en éliminant les REMs (Pom's N° 11) et en concaténant les lignes; vous pouvez aussi revoir votre technique de dessin (par exemple en utilisant une boucle FOR... NEXT qui lit les coordonnées en DATA et les écrit par HPLOT).

Troisième méthode: charger votre programme Basic au-dessus de la première page graphique utilisée; votre programme est alors localisé à une adresse inhabituelle mais vous disposerez de moins de place mémoire pour vos variables. Pour cela, exécutez le programme suivant avant votre propre programme :

```
10 POKE 104,64 : POKE
   16384,0:CLEAR:PRINT
   CHR$(4) "RUN progr."
```

Attention : cette dernière méthode dérange les pointeurs internes de l'Applesoft et il pourra vous être nécessaire de rebooter par la suite.

Il existe enfin une dernière technique plus élégante qui permet de faire "enjamber" un programme en Applesoft par-dessus les pages graphiques : voyez pour cela la page 63 du numéro 8 de Pom's ou le Recueil N°2 (p. 213-214) pour obtenir le listing du module en assembleur SPLITTER.

Dans le numéro 18 de Pom's, j'ai été très désappointé de m'apercevoir que le programme FENETRE ne fonctionnait pas correctement. En effet, la commande "& LET NV,VT,LN,NI(,IL)" n'est pas acceptée par mon Apple //e. Le message "ILLEGAL QUANTITY ERROR" apparaît.

Xavier Garcin
01630 St. Genis-Pouilly

Vous avez raison, il y a une erreur dans l'article, page 24. Il faut lire :
& LET NV,VT,HT,LN,NI(,IL)
avec :
NV nom d'une variable
VT tabulation verticale
HT tabulation horizontale
LN longueur de la barre inverse
NI nombre d'items
IL interligne (optionnel)

Par ailleurs, ce programme, décidément marqué par la malchance, avait subi un ennui de montage. Son listing était incomplet, et les lecteurs non abonnés aux disquettes ont dû attendre le n°19 pour pouvoir l'utiliser. Pardon ...

• *L'envoi de codes ASCII compris entre 0 et 127 peut être nécessaire (par exemple l'imprimante EPSON LQ1500 imprime obligatoirement en italique si elle reçoit des codes compris entre 128 et 255). Or la routine PRINT de l'Applesoft met automatiquement le 8ème bit à 1 et ne permet donc que d'envoyer des codes compris entre 128 et 255. Il en est de même d'AppleWriter. On peut généralement forcer le 8ème bit à 0 au niveau de la carte*

interface ou de l'imprimante en positionnant correctement les switches.
Mais alors on ne peut plus envoyer de codes compris entre 128 et 255, ce qui peut aussi être utile, mais de manière plus exceptionnelle, par exemple pour définir des caractères programmables. Le problème ayant été déplacé peut être résolu en utilisant une routine en langage machine telle que celle proposée par Apple Seedrin (Pom's n° 12), ou, plus simplement, par un POKE A,X où X est le code ASCII à envoyer (compris entre 0 et 255) et A est l'adresse correspondant à la sortie des données sur la carte interface. Elle dépend évidemment de la carte utilisée. Par exemple, pour la carte Supersérie Apple placée dans le slot 1, A est égal à 49304. Si l'imprimante n'a pas de buffer, ou si la liaison est particulièrement lente, il peut être nécessaire de prévoir une temporisation entre 2 POKEs successifs, pour éviter que des caractères se perdent.
Notons au passage que l'imprimante ImageWriter interprète correctement le 8ème bit lors des séquences de définition de caractères programmables, même lorsque les switches sont positionnés pour forcer le 8ème bit à 0.

- *Le 'ç' (ASCII 92) ne peut pas être utilisé dans les textes imprimés par APPLEWRITER (version][+) car il est interprété comme caractère de soulignement. Un lecteur aurait-il trouvé le moyen de changer le caractère de soulignement? En attendant, j'ai défini le 'ç' comme un caractère programmable (associé au code ASCII 106) de mon imprimante IMAGEWRITER et j'ai prévu un élément du glossaire qui insère dans mon texte la séquence des codes ASCII 27, 44, 106 ,27 ,36. Ainsi, par CTL-G suivi d'une seule touche, je passe en alphabet "utilisateur" (27,44), j'imprime le 'ç' (106) et je reviens au jeu de caractères normaux (27,36). On peut faire la même chose pour une autre imprimante.*

- *Encore un problème de 8ème bit: dans le n° 9 de Pom's, O. Herz propose une routine qui permet de récupérer dans un tableau de variables alphanumériques le contenu d'un catalogue de disquette DOS. En fait, on récupère dans ces variables des caractères dont le 8ème bit est forcé à 1. Ce n'est pas gênant dans le programme de démonstration, mais si l'on utilise cette routine dans un autre programme, il est impossible, par exemple, de comparer des noms de fichiers tirés du catalogue et des noms de fichiers entrés au clavier (qui sont en mémoire avec le 8ème bit égal à 0). La solution consiste à forcer à 0 le 8ème bit dans la routine d'O. Herz (AND #%01111111, en assembleur). La chaîne de caractères A$ de la ligne 10 devient alors: "300:38 48 84 FD A0 00 29 7F 91 FE E6 FE D0 02 E6 FF 68 A4 FD 60 A9 A0 60 N D823G" et le POKE 56,18 de la ligne 40 devient POKE 56,20.*

Jean-Claude Lengrand, 78000 Versailles

Lorsqu'on utilise un programme de gestion de données avec une disquette "fichier" initialisée selon une procédure normale, il est toujours ennuyeux de rencontrer, lors de l'enregistrement de nouvelles données, l'erreur "disquette pleine". L'appel d'une simple routine en langage machine, chargée à l'adresse $300, permet de connaître le nombre de secteurs encore libres. En enregistrant momentanément les données soit dans la carte langage, soit dans la mémoire de la carte 80 colonnes "étendue", initialisée respectivement en pseudo-disquette de 16K (voir Pom's 12, "Disque virtuel 16Ko"), ou 64K (voir Pom's 14, "Disque virtuel 64Ko"), et en faisant un appel, avant et après, à cette routine, il est possible de confronter l'occupation en secteurs de ces données avec le nombre de secteurs restés libres sur la disquette "fichier", et de décider par programme l'initialisation d'une autre disquette. Cette routine est tirée du programme "HELLO complet" (Pom's 6) et adaptée comme suit :

FRE.S

```
 1          ORG   $300
 2  VTOC    EQU   $B3F2
 3          JSR   $AFF7
 4          LDA   #$00
 5          STA   COUNT
 6          STA   COUNT+1
 7          LDY   #$8C
 8  NEXTS   LDA   VTOC,Y
 9  TESTS   BEQ   NEXTT
10          ASL
11          BCC   TESTS
12          PHP
13          INC   COUNT
14          BNE   OTHR
15          INC   COUNT+1
16  OTHR    PLP
17          BNE   TESTS+2
18  NEXTT   DEY
19          BNE   NEXTS
20          RTS
21  COUNT   DFB   2
22          END
```

FRE

```
0300- 20 F7 AF A9 00 8D 25 03
0308- 8D 26 03 A0 8C B9 F2 B3
0310- F0 0F 0A 90 FB 08 EE 25
0318- 03 D0 03 EE 26 03 28 D0
0320- F1 88 D0 E9 60
```

Sauver cette routine comme suit :

```
BSAVE LIBRE,A$300,L$25
```

En début de programme de gestion, mettre l'instruction :

```
PRINT CHR$(4)"BLOAD
LIBRE,S6,D1"
```

Pour connaître le nombre de secteurs libres "SL", "s" étant le numéro du slot et "d" le numéro du drive, insérer cette ligne d'instructions :

```
POKE 43626,s:
POKE 43624,d:
CALL 768:
SL=PEEK(805)+
PEEK(806)*256
```

Pour initialiser une nouvelle disquette "fichier" avec 526 secteurs, insérer cette ligne d'instructions :

```
POKE 46922,96:
POKE 46923,234:
POKE 44723,4:
PRINT CHR$(4)"INIT
XX,S6,D2":
PRINT CHR$(4)
"DELETE XX"
```

Thierry Haurie - Via Napoli , 2b 21027 ISPRA (VA) - Italie

Micro-Informations

Jean-Michel Gourévitch

Pour les aficionados d'Apple, la nouvelle a résonné comme un déchirement : Steve Jobs s'en va, la société qu'il a créée lui intente un procès. Après Steve Wozniak, voici que part l'autre père fondateur d'Apple, et un peu de la micro-informatique. On aura beau dire que la croissance d'une société a ses raisons, qu'il faut, un jour, trancher le cordon ombilical, il n'empêche : les Applemaniaques en auront comme un pincement au cœur...

Pas de quoi paniquer, cependant, dans l'immédiat : Jobs a quitté Apple avec des idées et quelques ingénieurs. On entendra à coup sûr reparler de lui. Quant à Apple, la société à la pomme ne manque pas de projets dans ses cartons pour les trois à quatre années à venir. Après, on verra...

Parmi ces projets, on trouvera de quoi moderniser l'Apple // et muscler le Macintosh.

Moderniser l'Apple //

Priorité chez Apple : la remise au goût – et à la technique – du jour de l'Apple //. Si, en France, ses ventes ont un peu faibli en octobre, aux Etats-Unis, il continue de résister vaillamment. Pour preuve, le contrat passé avec l'administration des postes de ce pays qui en a acheté en septembre 700 d'un coup (s'ajoutant aux 1400 acquis au début 1985). Au Sicob, Apple France a annoncé officiellement les nouveautés de l'Apple // qui avaient été présentées aux lecteurs de Pom's dans le numéro 20 : lecteurs Unidisk 3.5 de 800 Ko, carte d'extension de mémoire de 1 Méga-octet, imprimante Imagewriter II. Voici donc l'Apple // officiellement nanti de davantage de mémoire que le Macintosh.

Et cela n'est qu'un début : les fabricants de cartes d'extension s'agitent. L'un d'eux, **Coit Valley Computers**, a trouvé moyen de caser 1536 Ko (plus d'un Mo et demi !) sur sa **MultiRAM** //e (utilisable seulement sur l'Apple //e). Peuvent l'utiliser tous les programmes écrits aux spécifications d'Apple, et notamment Appleworks : la carte est livrée avec un "patch" pour étendre la capacité d'Appleworks, un RAM disk permettant d'utiliser la carte comme disque virtuel, ainsi qu'un test de mémoires; un bus de la carte permet d'y relier un coprocesseur 16 bits pour accélérer les traitements). Son prix est presque raisonnable : 699 dollars.

Une fois la mémoire de l'Apple // "gonflée", encore faut-il l'utiliser, et avec une rapidité suffisante. Après avoir sorti MouseDesk, son "Finder" utilitaire style Mac, à fenêtres et menus déroulants, **VersionSoft** travaillerait actuellement sur un "**Switcher**" pour l'Apple // (le Switcher permet de partitionner la mémoire de l'ordinateur, et d'y installer simultanément plusieurs programmes qui peuvent communiquer entre eux). Avec 1,5 Mo, il y a de quoi nicher pas mal d'applications et de fichiers en mémoire.

Reste le second problème : la rapidité. Le "vieux" processeur 6502 de l'Apple // n'avait pas été conçu pour subir de telles épreuves ni gérer de telles mémoires. Il faudra donc incorporer à l'Apple // un nouveau processeur. Peut-être le 68516, qui se substituerait au 6502.

Une autre solution serait possible : installer sur une carte (en coprocesseur, donc) le tout nouveau 68020 de Motorola. Quel est encore celui-là ? Tout simplement la dernière version du processeur 68000 qui équipe... le Macintosh, plus sophistiquée : bus 32 bits, capable d'adresser 4 Giga-octets de mémoire, et surtout encore plus rapide (12,5 Mhz contre 7,83 Mhz actuellement). Rêvons un peu : l'Apple // muni du même format de disquettes et du même processeur, et qui simulerait le fonctionnement du Mac ? Ce serait trop beau pour être vrai...

Muscler le Macintosh

En tout cas, on parle déjà de ce processeur pour équiper le Macintosh. Ou plutôt son successeur, car il est improbable que l'on puisse mettre à niveau les Macs d'aujourd'hui. Avec ce processeur, disposer de deux à quatre Mo de mémoire vive, et améliorer rapidité et gestion d'écran deviendrait possible.

Le problème actuel du Macintosh, comme l'ont remarqué ses utilisateurs, c'est qu'on atteint trop vite les limites de l'ordinateur. L'utiliser avec un système un peu copieux (enflé notamment de polices de caractères et d'accessoires de bureau) et des programmes "mémorivores" comme Jazz exige désormais une greffe de mémoire et un disque dur. Et bien que les prix du Macintosh aient été réduits en septembre (de quelques 6 000,00 F HT pour le 512 Ko), l'addition atteint encore des sommets vertigineux. A la mi-octobre, il fallait compter environ 16 000,00 F pour le nouveau disque dur d'Apple (pas encore disponible), et un peu plus de 5 000,00 F pour le passage à 1 024 Ko (greffe de 512 Ko supplémentaires). Aux Etats-Unis, Beck Tech propose cette transformation à 1 024 Ko pour $549, et offre aussi une extension portant la mémoire vive totale à deux Mo. Qui dit mieux ?

Attention cependant : pour franchir la frontière des 512Ko, et utiliser cette mémoire supplémentaire directement (sans passer par un RAM disk), il est nécessaire de modifier la ROM du Macintosh, de façon à déplacer la zone de mémoire vive gérant l'écran du Mac. Une modification qui n'est pas sans danger et qui peut poser des problèmes de garantie.

Voici une des raisons pour lesquelles les utilisateurs attendent avec impatience la nouvelle version de la ROM (128 Ko) qu'Apple préparerait pour le Macintosh. On risque hélas de l'attendre jusqu'à 1986. De même que le lecteur de disquettes double face. Il sera d'ailleurs peut-être remplacé par un lecteur de plus grande capacité.

C'est en janvier prochain, lors de l'Assemblée générale des actionnaires Apple que toutes ces nouveautés devraient être annoncées. En attendant, il faudra faire avec ou, pour être plus précis, sans...

Apple //e : de plus en plus de mémoire

L'Apple // suscite des perfectionnements qui augmentent notablement sa puissance.

Côté stockage, voici le premier disque dur à installer *à l'intérieur* de l'Apple //. Surprise : il est français. D'une capacité de 10 Mo, ce disque dur est conçu par **Gerb Electronique**, une société de Saint Nazaire. Comment loger un disque dur sous le capot de l'Apple // ? Bonne question, car il n'y a guère de place. Eh bien, on commence par retirer le bloc d'alimentation, et on lui substitue le Mega-Core, qui contient à la fois une alimentation de dimensions plus réduites, et un disque dur de 3 pouces et demi d'une capacité de 10 Mo. La carte de contrôle du disque s'installe dans un des slots de l'Apple; elle peut gérer 64 Mo. Le logiciel permettrait de gérer le disque dur avec le système ProDOS, ce qui est normal, mais aussi avec le Pascal, le CP/M et le Dos 3.3, ce qui l'est moins. Prix : 18 000,00 F.

Côté mémoire vive (MEV), voici la **Kache Board d'OKS**. Une carte "intelligente" qui s'installe dans l'un des slots 4, 5, 6 ou 7. Elle contient 256 Ko, et conserve en mémoire seulement les données les plus souvent utilisées; elle procède automatiquement à des sauvegardes régulières sur le disque dur, et réduit ainsi les temps d'accès. Un complément utile d'un disque dur pour $695.

Côté processeur, voici une nouvelle carte incorporant un microprocesseur 65C816 (16 bits) : la **MultiRAM IIEX de Checkmate**, qui comprend aussi une extension de mémoire. On enlève le processeur et la ROM de gestion de mémoire du chassis de l'Apple //, et on les enfiche sur la carte. Cette même carte inclut le **Max Operating System de Micromagic** (décrit dans le précédent numéro de Pom's). Prix de la carte : $200. Pour le moment, le hic, c'est qu'on ne trouve guère de programmes développés pour le 65C816. Et l'année prochaine ? Si l'Apple // se met au 16 bits...

Apple //c : de plus en plus de mémoire

La carte d'extension de mémoire d'Apple ne s'adresse qu'au IIe, mais les fabricants de cartes ont aussi pensé au IIc.

Applied Engineering propose ainsi sa **Z RAM** pour le IIc. Elle s'installe à l'intérieur du //c et porte la mémoire vive à 640 Ko. Utilisables en disque virtuel, ou directement par certains programmes comme AppleWorks. Utilisables aussi avec le Pascal, le DOS 3.3, CP/M et le ProDOS. En prime, la carte contient un processeur Z-80 permettant, en effet, au //c de faire tourner tous les programmes en CP/M (DBase II, Wordstar, etc.). Prix : $549 pour les 512 Ko.

Checkmate Technology offre la **MultiRAM C** avec également 512 Ko pour $429, mais sans le Z-80.

Toujours pour le IIc, **Applied Engineering** vend aussi ($79) une horloge qui se branche sur le port série.

Des classiques

Sur le front des logiciels, voici d'abord des classiques qui nous reviennent de divers bords.

Word Perfect de **SSI**, célèbre car il s'agit d'un des meilleurs traitements de texte de l'IBM PC, vient de se convertir à l'Apple //. Il dispose de tout ce qu'on peut attendre d'un traitement de texte : visualisation à l'écran du texte tel qu'il sera imprimé, remplacement des mots, déplacements de paragraphes et de pages, correction orthographique (hélas, en anglais seulement), et il est configuré pour de nombreuses imprimantes. Prix : $179.

Wordstar est aussi utilisable sur l'Apple //, à condition de disposer d'une carte d'extension CP/M. Voici quelque temps, son éditeur **MicroPro** proposait un ensemble comprenant Wordstar et la carte. Cet ensemble, composé de Wordstar, Starindex, Spellstar, Report Star, Mailmerge, Datastar et Calcstar, ainsi que d'une carte CP/M fonctionnant à 6 Mhz, est bradé aujourd'hui par **Total Access** à 299 dollars. Un prix défiant toute concurrence.

Quant à **Software Publishing**, l'éditeur des célèbres PFS File et Write, il vient d'améliorer son tableur **PFS Plan**. Celui-ci gère maintenant en mémoire 70 rangs de 36 colonnes. Utilisation particulièrement facile, prix : $125.

Des accessoires de bureau

Monitor, par **Dogstar Software**, est un de ces programmes résidant en mémoire dont on a l'habitude sur l'IBM PC. Il s'installe derrière tout programme écrit en Pascal et permet, en pressant sur une touche, de disposer d'une calculatrice, du catalogue des fichiers, de fonctions de gestion de mémoire et de disquettes. Le programme offre même un mini assembleur-désassembleur. Prix : 49 dollars.

On connaissait les accessoires de bureau du Macintosh, en voici pour votre Apple // préféré : les **AppleWorks Accessories**. Ils s'installent dans Appleworks et ont été conçus par Pinpoint. On y trouve un carnet de notes, une calculatrice, un agenda, un composeur automatique de numéros de téléphone, et enfin la possibilité d'inclure des dessins dans des documents ou d'imprimer des adresses sur des enveloppes. Le tout pour seulement $69.

D'autres accessoires de bureau, avec **Jeeves** de **PBI**. Une seule pression sur une touche du clavier, et l'exécution d'AppleWorks est suspendue; on peut alors utiliser un calendrier, un composeur de numéros de téléphone avec annuaire, une calculatrice à quatre fonctions, un bloc-notes ou un réveil. Prix : $50.

Des compatibles

On les croyait disparus avec l'arrivée des Apple //e et //c : revoici les ordinateurs compatibles. Aux Etats-Unis, **Franklin** sort l'**Ace 2000** ($699) compatible avec le //e et le //c. Grâce à un nouveau DOS : le FDos.

Boîtes métalliques, double haute résolution graphique, caractères souris, claviers numériques. Ces machines ne violeraient pas le copyright de la ROM d'Apple.

En France, certaines boutiques commencent à offrir des compatibles "made in Taïwan"; certains proposent même d'acheter les ROMs chez Apple. Attention à ces matériels vendus à des prix intéressants, mais dont la légalité est douteuse. En tout cas, le fait qu'on continue à copier l'Apple // montre qu'il est loin d'être enterré. Les lecteurs de Pom's en doutaient-ils ?

Pour le Macintosh : des langages

Par la facilité de son apprentissage, le Macintosh se révèle un outil de choix pour l'application des systèmes experts et des langages d'intelligence artificielle. Ces langages se multiplient à présent pour le Mac.

Tout d'abord, la version Mac du Prolog II. Bénéficiant de l'environnement Macintosh (fenêtres, souris, etc.). Ce langage a été défini à partir du Prolog II conçu par le GIA (Groupe d'intelligence artificielle de Luminy, Université d'Aix Marseille II). Le Prolog II est distribué par Apple et vendu au prix de 4 744,00 F TTC.

Ensuite, voici le **MacExpert** de **MindSoft**. Un système expert "d'ordre zéro", capable de créer un système résolvant dilemmes ou alternatives, en matière de diagnostic, de classification, de planification financière, etc. Ce langage est basé sur le calcul des propositions et ses bases de connaissance sont composées de règles construites à partir de phrases déclaratives. Ce système autorise 250 règles avec un Mac de 128 Ko et jusqu'à 1 000 règles avec un 512 Ko.

Autre système expert, **Nexpert**, développé par trois français (Alain Rappaport, Jean-Marie Chauvet et Patrick Perez) a fait l'admiration des américains. Ce programme – vendu quelques $5000 – accomplit les mêmes tâches que d'autres vendus plus de $100 000. Utilisant l'interface graphique du Macintosh, sa structure de connaissance est automatiquement construite, visualisée et mise à jour par le système.

Enfin, **Experlogo**, un produit de **Turgeon**, est un Logo pour le Mac permettant la réalisation de graphiques.

Communication...

... grâce au modem développé par **Hello Informatique**. Il numérote, se connecte et déconnecte automatiquement. Voici pourquoi ce modem ne comporte aucun bouton, sauf un de réinitialisation exceptionnelle du processeur. Ce véritable micro de communication comporte, comme l'Apple //c, un micro-processeur 65C02. Il peut donc être programmé. On peut développer un programme de communication et le télécharger dans le modem. Résultat : ce modem peut transformer l'ordinateur en serveur, il peut aussi servir de répondeur automatique, enregistrant automatiquement les messages des terminaux, ou des minitels qui l'appellent, et ce sans mobiliser l'ordinateur. Il peut aussi servir à crypter ou décrypter les données échangées. Bien évidemment, il fonctionne avec Mac Tell, MacTerminal, Version Com ou ASCII Express, et permet d'accéder aux serveurs à 1.200 bauds. Le tout pour moins de 5.000 francs.

Apple France offrira aussi deux nouveaux modems : l'un en bas de gamme pour utilisation à 300 bauds, à faible coût. L'autre, plus évolué, à 1 200 bauds.

Des utilitaires

Macdisk Catalog, de **New Canaan Micro Code**, permet d'établir un catalogue de vos disquettes et des fichiers qui s'y trouvent, avec un maximum de 1092 fichiers par catalogue. Il permet aussi d'imprimer des cartes de référence. Prix : $40.

Dream of the Phoenix propose des disquettes vendues 40 dollars, et remplies à ras bord d'utilitaires utiles. L'une contient un calendrier, une autre un programme de communication permettant d'établir un bulletin. Les **Quick and Dirty Utilities** (en deux volumes) comprennent des configurations de terminaux s'installant dans le menu Pomme, des grapheurs, etc.

Des accessoires de bureau

Le menu "Pomme" du Macintosh est devenu le refuge favori des accessoires de bureau. Beaucoup sont distribués par les "Bulletins boards" américains. On y trouve notamment un compteur de mots (incapable cependant de compter les mots dans MacWrite), une horloge stylisée au nom parlant ("Big Ben"), des chronos décomptant le temps, un programme permettant d'apprendre le Morse. Et même un mini "grapheur" permettant de réaliser à loisir camemberts, courbes, etc. en n'occupant que 18 Ko.
Attention cependant aux incompatibilités : ainsi **MacTracks**, qui permet de créer des macros (séquences de commandes automatiques) dans un programme ne fonctionne pas avec le précieux **Art Grabber** (pour ouvrir un dessin MacPaint et l'incorporer dans n'importe quel programme). On risque souvent de voir apparaître une bombe en appelant un de ces accessoires...

Calculator Construction Set de **Dubl Click** permet de construire soi-même une calculatrice personnalisée et de l'inclure dans le menu Pomme. Prix : $99.

Et voici enfin, pour Macintosh, **Sidekick**, le célèbre programme développé par le non moins célèbre **Philippe Kahn** – patron français de Borland – et qui a fait un malheur sur l'IBM PC. La version Macintosh comprend une partie s'intégrant dans le menu "Pomme" avec un composeur de numéros de téléphone (il manque l'interface pour le relier au réseau français), un mini programme de communications invoquable depuis n'importe quelle application, une calculatrice plus perfectionnée que celle du Mac (permettant notamment d'imprimer la "bande" de papier des calculs), un "spooler" pour imprimer un texte tout en continuant à travailler avec le Mac (imprimer seulement des textes non formatés), une horloge analogique et un calepin plus évolué que celui d'Apple. Avec ce calepin, on peut ouvrir n'importe quel fichier à l'intérieur d'un programme, prendre des notes, procéder à du Couper/ Coller entre programmes. Cet instrument est génial et permet notamment de disposer de fenêtres dans MacWrite, alors qu'elles n'existent naturellement que dans Word. Egalement regroupées dans le menu "Pomme", des mini-feuilles de calculs permettant de calculer des frais, de noter une liste de choses à faire, etc.

Quant à **Sidekick** lui-même, il s'agit d'un programme permettant d'établir un annuaire et, en cliquant sur un nom, de composer automatiquement le numéro de

téléphone (à condition de disposer de l'interface nécessaire), de calculer automatiquement le coût de la communication et de prendre des notes. Un de ces programmes dont les Américains raffolent, qui a aussi soulevé l'enthousiasme de J.-L. Gassée, l'ex-patron d'Apple France; il souffre cependant d'un handicap : il occupe à lui seul une partie de la disquette, et son système est mémorivore. Pour l'utiliser, il faut :

1) deux lecteurs;
2) se servir d'une disquette pour le seul système;
3) installer les applications et fichiers sur la seconde.

Pour un fonctionnement optimal – et cette règle vaut pour tous les accessoires de bureau – il vaut mieux disposer d'une mémoire de 1 Mo et d'un disque dur ...

Des jeux

Pinball Construction Set, d'Electronic Arts, ce flipper en kit qui fit les beaux jours de l'Apple //, est maintenant disponible pour le Macintosh. On peut même dessiner ses décors avec MacPaint.

Fokker Triplane est l'un des premiers simulateurs de vol pour le Mac, simulant, comme son nom l'indique, un triplan Fokker de la première guerre mondiale.

De la musique

De nouveaux programmes rendent encore plus musicaux aussi bien l'Apple // que le Mac.

MUSE (MIDI Users Sequencer Editor), de **Rolandcorp**, permet de composer et jouer de la musique sur l'Apple // en utilisant l'interface MIDI, qui est devenue le nouveau standard en matière de musique d'ordinateurs. Possibilités : huit pistes indépendantes, correction automatique des erreurs rythmiques et utilisation possible avec un joystick. Prix : $150.

Total Music, de Southwork Music Systems, permet d'interfacer un Mac au standard MIDI. On peut enregistrer jusqu'à deux instruments MIDI à la fois. Prix : $489.

Mega Track, de Musicworks, permet d'enregistrer directement depuis des instrumens MIDI. Pour 150 dollars. A noter que tous ces systèmes permettent d'inclure les signaux de codes horaires (SMPTE) rendant le Mac compatible avec les systèmes professionnels de vidéo et de cinéma.

Enfin, **Studio Mac de Creative Solutions**, utilisant aussi l'interface MIDI, permet de simuler un clavier de synthétiseur à 8 voix pour $595.

Le Mac devient ainsi peu à peu un instrument de musique étonnant.

Adresses

Applied Engineering
PO Box 798
Carrollton, Texas 75006

Beck Tech
Berkeley – California

Checkmate Technology
509 South Rockford Drive
Tempe, Arizona 85281-3021

Coit Valley Computers
14055 Waterfall Way
Dallas, Texas 75240

Creative Solutions
4701 Randolph Rd Suite
12 Rockville, MD 20852

Dogstar Software
PO Box 302
Bloomington, IN 47402

Dream of the Phoenix
PO Box 10273
Jacksonville, Florida 32247

Dubl Click Software
18201 Gresham Street
Northridge, CA 91325

Fraciel
42, rue des Prébendes
37000 Tours

Gerb Electronique
ZI de Brais – 44600 St Nazaire

Hello Informatique
1, rue de Metz – 75010 Paris

JCR
58, rue Notre Dame de Lorette
75009 Paris

MindSoft
11, bd. Edgard Quinet
75014 Paris

Musicworks
18 Haviland
Boston, MA 02115

New Canaan Micro Code
136 Beech Rd.
New Canaan, CT 06480

OKS
75 Tahlequah Trail – Dept.11
Springboro, Ohio 45066-1154

PBI Software
1155B-H Chess Drive
Foster City, CA 94404

Pinpoint Publishing
Tél : (415) 5301726 (USA)

Rolandcorp US
7200 Dominion Circle
Los Angeles, CA 90040

Software Publishing
1901 Landings Drive
Mountain View, CA 94043

SouthWorth Music Systems
Box 275, R.R.1
Harvard, MA O1451

SSI
288 West Center Street
Orem, Utah 84057

Total Access
PO Box 764246
Redbird Airport – Hangar #8
Dallas, TX 75376-4246

Turgeon
17, rue Sextius Michel
75015 Paris

Version Soft
66, rue Castagnary – 75015 Paris

Entretien avec Jean-Louis Gassée

Jean-Louis Gassée, de passage en France pour la promotion de son livre "La troisième pomme" (voir bibliographie), a accordé une interview à Pom's, dont nous vous faisons partager quelques extraits. Nous n'avons hélas pas la place de retranscrire ici l'équivalent d'une heure de propos à bâtons rompus. Nous avons fait cependant notre possible pour respecter l'ensemble de l'interview, malgré les indispensables coupures.

> Quelle est aujourd'hui la stratégie d'Apple ?

JLG : La stratégie d'Apple est en cinq points :

• tout d'abord, de développer deux lignes de produits, Apple II et Macintosh, avec des modèles compacts et extensibles.

• une ligne de périphériques communs, en fait, des standards de connection qui feront, à terme, que si par exemple on a un disque dur, il se connectera sur tous les appareils de la gamme ; on simplifie ainsi la vie de nos clients, de nos revendeurs et des auteurs de logiciel.

• mettre à la disposition des utilisateurs un interface matériel et logiciel pour la communication qui leur permettent de disposer d'une interface de communication normalisée, homogénéisée. C'est un problème technique considérable mais je pense que, si on le résoud, on offrira des outils extrêmement compétitifs sur le marché.

• faire en sorte que nos clients eux-mêmes puissent être compétitifs dans un monde à fournisseurs multiples, qu'ils puissent lire les fichiers établis avec les machines d'autres marques.

• le dernier point est un peu plus à long terme, mais il en faut. C'est de faire des recherches sur les architectures du futur. Tout simplement parce que la vocation d'Apple, c'est d'intercepter l'innovation et de l'industrialiser.

Aujourd'hui, il y a tout un tas de recherches qui se font en logiciel et qui vont aboutir. Ce qu'on essaie de faire avec les architectures du futur, c'est d'être prêts à les intercepter avec des machines qui seront par exemple dix fois aussi puissantes qu'un Macintosh, plus puissantes d'ailleurs que pas mal de super-minis d'aujourd'hui. C'est à long terme, mais on pense avoir quelques idées sur la question. On a un budget recherche et développement pour 1986 qui dépasse 100 millions de dollars, alors ...

> Avec l'extension de la gamme Apple II comme de celle du Macintosh, n'y a-t-il pas un risque sérieux d'augmenter les problèmes de compatibilité ?

C'est l'éternel problème. Ma préférence, quand j'ai le choix, va vers le maintien de la compatibilité plutôt que la rupture. Je pense que cela ne s'est pas trop mal passé avec le IIc. On a parfois des surprises avec la compatibilité : on se dit après coup "Mais comment je n'y ai pas pensé...". Nous faisons travailler beaucoup de testeurs.

Ceci dit, l'Apple II va entrer en 86 dans sa dixième année : cela a forcément changé entre les machines d'aujourd'hui et celle du début. La meilleure stratégie d'un constructeur, c'est d'organiser quand même une certaine dose d'incompatibilité : quand vous voulez apporter des progrès, si vous n'abandonnez pas quelque chose de temps en temps, vous allez faire un monstre qui marchera avec toutes les cartes depuis le début de l'humanité et tous les logiciels.

> Et l'avenir ?

JLG : Il y a deux voies qui nous intéressent à très long terme.

La première, c'est comment utiliser la technologie des calculateurs à jeu d'instructions réduit, des machines qui, au lieu d'avoir des jeux d'instructions très riches, auront un parti pris très différent : sur une même surface de silicium, on va mettre une logique beaucoup plus simple et une énorme quantité de registres. Avec des centaines de registres, quand on programme cela efficacement, on peut avoir des vitesses de calcul – avec la technologie actuelle, le silicium – extraordinaires. Si on projette cela dans trois à cinq ans, c'est une des voies.

L'autre voie, c'est d'explorer aussi les architectures multicircuits. C'est très compliqué parce que, dans ce domaine-là, un plus un ne fait pas deux.

Bon de commande

Disquettes

HAIFA source	(cf. Pom's n° 5)	.....	à 55,00 F	.....
H-BASIC	(cf. Pom's n° 8)	.....	à 150,00 F	.....
MUSIC	(cf. Pom's n° 10)	.....	à 80,00 F	.....
DISK-MANAGER	(cf. Pom's n° 11)	.....	à 450,00 F	.....
BASICIUM	(cf. Pom's n° 13)	.....	à 150,00 F	.....
E.P.E.	(cf. Pom's n° 15)	.....	à 150,00 F	.....
PASCAL	(cf. Pom's n° 15)	.....	à 80,00 F	.....
MAX (Moniteur étendu)	(cf. Pom's n° 18)	.....	à 150,00 F	.....
DOMINOS	(cf. Pom's n° 19)	.....	à 80,00 F	.....
MACASTUCES	(cf. Page 68)	.....	à 200,00 F	.....
P-FORMAT][, P-POLICE][	(cf. Page 33)	.....	à 200,00 F	.....
COGO	(cf. Page 29)	.....	à 150,00 F	.....

Recueils

N°1, recueil des revues 1 à 4	.....	à 140,00 F	.....
Disquettes d'accompagnement 1 à 4	.....	à 150,00 F	.....
N°2, recueil des revues 5 à 8	.....	à 140,00 F	.....
Disquettes d'accompagnement 5 à 8	.....	à 190,00 F	.....

Anciens numéros

Revues 4 7 8	.....	à 35,00 F	.....
Revues 9 10 11 12 13 14 15 16 17 18 19 20 21	.....	à 40,00 F	.....
Disquettes Apple II, //e, //c 1/2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21	.....	à 55,00 F	.....
Disquettes Macintosh 14/15/16 groupées	.....	à 150,00 F	.....
17 18 19 20 21	.....	à 80,00 F	.....

Abonnements

Pour 6 numéros à partir du n° ☐

Abonnement à la revue seule	.....	à 200,00 F	.....
Abonnement revue + disquettes Apple II, //e, //c	.....	à 480,00 F	.....
Abonnement revue + disquettes Macintosh	.....	à 600,00 F	.....

Total TTC : _____

Supplément avion hors CEE : 15,00F par numéro et/ou disquette : _____

Montant du règlement : _____

Envoyez ce bon et votre règlement à : EDITIONS MEV, 64 rue des Chantiers 78000 VERSAILLES

Nom : .....

Adresse : .....

Imprimerie Rosay, 94300 Vincennes. Imprimé en France. Dépôt légal : Novembre 1985. N° 5078.

4e DIMENSION

Une base de données relationnelles qui propulse Macintosh dans un univers surpuissant et jamais atteint.

Demandez à Macintosh ce que vous n'auriez jamais osé lui demander... c'est tout simple.

ACI

Analyses Conseils Informations
38 avenue Hoche 75008 Paris
tél. (1) 359.89.55

MULTIFICHIER RELATIONNEL
- cent fichiers en ligne
- nombre illimité de sous-structures
- liens graphiques entre les fichiers
- relations générées automatiquement

RICHESSE FONCTIONNELLE
- nombre illimité de critères, d'index, de formats de saisie, de consultation, d'impression
- recherche multicritère illimitée

GÉNÉRATEUR D'APPLICATIONS SIMPLE ET PUISSANT
- langage de programmation en organigramme
- menus personnalisés

patrick chédal C&C

APPLE CHEZ P.S.I.

35 TITRES APPLE AU CATALOGUE P.S.I.

Apple, Logique et systèmes experts.
par René DESCAMPS.
224 pages - 120,00 FF.
Pour résoudre des énigmes logiques et bâtir des raisonnements.

Basic Plus 80 routines sur Apple II,
par Michel MARTIN.
144 pages - 95,00 FF.
Pour muscler votre machine.

Exploitation d'enquêtes sur Apple II et IBM/PC,
par J.-F. GRIMMER.
176 pages - 120,00 FF.
Le traitement d'enquêtes et l'analyse des données.

Diététique sur Apple II,
224 pages - 120,00 FF.
Mangez équilibré grâce à votre ordinateur.

Apple en famille,
par J.-F. SEHAN.
232 pages - 120,00 FF.
40 programmes utilisés en Basic.

Super Jeux Apple,
256 pages - 120,00 FF.
50 programmes de jeux en Basic sur Apple.

Apple, modems et serveurs,
par Alain MARIATTE.
224 pages - 130,00 FF.
Apprenez à communiquer avec votre Apple.

50 modèles Multiplan pour gérer sur Apple et IBM/PC,
par P. et F. GYSEL.
50 applications concrètes intéressant de nombreux aspects de la vie de l'entreprise.

La programmation des jeux d'Arcades sur Apple IIe et IIc,
par Jean-Luc FISCHER.
340 pages - 140,00 FF.
Apprenez à créer vos programmes.

En vente en librairie, en boutique informatique et auprès de P.S.I. Diffusion

B.P. 86 - 77402 LAGNY-SUR-MARNE CEDEX
Tél. : (6) 60 06 44 35 - Télex : PSIDIF 600978 F